

Спицина И. А.
Аксенов К.А.

**Метод поддержки принятия
решений при разработке
информационных систем
на основе мультиагентного
подхода**

Монография

Министерство науки и высшего образования Российской Федерации
ФГАОУ ВО «Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина»
ФГБОУ ВО «Уральский государственный педагогический университет»

Спицина И. А., Аксенов К.А.

**Метод поддержки принятия решений
при разработке информационных систем
на основе мультиагентного подхода**

Екатеринбург 2018

УДК 004
ББК 3972.53
С72

Рецензенты:

Снежко В. Л., доктор технических наук, профессор, заведующий кафедрой «Информационные технологии в строительстве» Института мелиорации, водного хозяйства и строительства, ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К. А. Тимирязева»

Бегунов Н. А., кандидат экономических наук, начальник отдела математического моделирования Единого аналитического центра ПАО КБ «Уральский банк реконструкции и развития»

Спицина, И. А.

А41 Метод поддержки принятия решений при разработке информационных систем на основе мультиагентного подхода [Текст] : монография / И. А. Спицина, К. А. Аксенов. – Екатеринбург : УрГПУ, 2018. – 156 с.

ISBN 978-5-7186-1078-9

Монография посвящена вопросам разработки мультиагентного метода анализа и синтеза информационных систем с использованием современных средств вычислительной техники, информационных технологий и визуального программирования. Проектирование и разработка информационных систем является достаточно сложной и творческой задачей. Для ее решения применяются методы принятия решений, системный анализ, теория проектирования информационных систем, имитационное и экспертное моделирование, методологии структурного и объектно-ориентированного анализа и проектирования, мультиагентный подход.

УДК 004
ББК 3972.53

ISBN 978-5-7186-1078-9

© Спицина И. А., Аксенов К. А., 2018
© ФГБОУ ВО «УрГПУ», 2018

ОГЛАВЛЕНИЕ

Введение	5
1. Процесс системного анализа при разработке информационных систем	8
1.1. Этапы системного анализа при разработке информационных систем.....	8
1.1.1. Организационно-технические системы.....	8
1.1.2. Моделирование автоматизируемых процессов.....	10
1.1.3. Реинжиниринг бизнес-процессов.....	15
1.2. Риски, связанные с разработкой информационных систем, и пути их снижения.....	16
1.3. Методологические и теоретические основы поддержки принятия решений, моделирования и разработки информационных систем.....	18
1.3.1. Применение имитационного моделирования при разработке программного обеспечения.....	18
1.3.2. Использование систем искусственного интеллекта при разработке информационных систем.....	20
1.3.3. Применение мультиагентного подхода.....	20
1.4. Обзор и сравнительный анализ существующих систем поддержки принятия решений в области разработки информационных систем (CASE-средств).....	23
1.4.1. Классификация CASE-средств.....	23
1.4.2. Описание CASE-средств.....	24
1.4.3. Критерии сравнения функциональных возможностей CASE-средств.....	31
1.4.4. Сравнительный анализ CASE-средств.....	31
2. Метод поддержки принятия решений при разработке информационных систем для предметной области мультиагентных процессов преобразования ресурсов	34
2.1. Требования к модели и методу поддержки принятия решений при разработке информационных систем.....	34
2.2. Выбор модели представления бизнес-процессов.....	35
2.3. Выбор модели представления знаний.....	37
2.4. Построение модели разработки информационной системы.....	41
2.4.1. Концептуальная модель предметной области мультиагентных процессов преобразования ресурсов.....	41
2.4.2. Концептуальная модель предметной области информационных систем...	49
2.5. Метод разработки информационных систем.....	54
2.6. Анализ задержек в синхронной информационной системе.....	80
2.7. Методика оценки эффективности работы метода разработки информационных систем.....	84
3. CASE-средство BPsim.SD	87
3.1. Функциональные возможности пакета Bpsim.SD.....	87
3.2. Описание CASE-средства Bpsim.SD.....	88
3.2.1. Общая структура CASE-средства Bpsim.SD.....	88
3.2.2. Создание диаграммы DFD.....	90
3.2.3. Создание диаграммы прецедентов.....	94
3.2.4. Создание диаграммы последовательности.....	96
3.2.5. Создание диаграммы классов.....	100
3.2.6. Подсистема моделирования пользовательского интерфейса.....	104
3.3. Описание агента интеграции Bpsim.MAS и Bpsim.SD.....	108
3.4. Методика использования пакета Bpsim.....	109

4. Применение СППР Bpsim.SD при разработке информационных систем.....	113
4.1. Проект по анализу бизнес-процессов и разработке технического задания на единую информационную систему вуза УГТУ-УПИ.....	113
4.1.1. Процесс «Ход сессии».....	114
4.1.2. Процесс «Движение контингента».....	116
4.1.3. Оценка эффективности внедрения модуля «Движение контингента».....	121
4.2. Разработка дополнительных модулей для системы ведения реестров акционеров «Вереком-2».....	124
4.2.1. Регистрация входящих документов.....	124
4.2.2. Модуль «Трансфер-агентский обмен».....	128
4.2.3. Система «Электронный документооборот».....	131
4.3. Экспериментальные оценки.....	134
Заключение.....	140
Список сокращений и условных обозначений.....	143
Список литературы.....	144

ВВЕДЕНИЕ

Данная работа посвящена вопросам разработки метода поддержки принятия решений при проектировании информационных систем (ИС) на основе мультиагентного подхода. Достаточно часто проекты по созданию информационных систем заканчиваются неудачно. На процесс разработки оказывают влияние различные факторы. Один из них – это искажение и потеря информации о разрабатываемой ИС и особенностях бизнес-процессов при передаче ее от специалистов в предметной области к разработчикам. Основным источником этих знаний является техническое задание (ТЗ), однако оно недостаточно полно отражает данные о процессах согласования и принятия решений в автоматизируемой предметной области и ожидания пользователей от новой системы.

Успешность разработки ИС можно повысить благодаря использованию в процессе проектирования проработанного методологического подхода на основе комбинации существующих в настоящее время методов. Структурный подход (IDEF0, DFD) позволяет описать разрабатываемую систему в виде иерархии взаимосвязанных функций. Такое представление понятно аналитику и пользователю. Для анализа узких мест и динамических характеристик используется имитационное моделирование. При описании модели разрабатываемой системы с точки зрения разработчика применяют объектно-ориентированный подход (язык UML). Экспертные системы закрывают вопросы, связанные с описанием знаний и сценариев принятия решений. Мультиагентные системы (МАС) позволяют автоматизировать процессы согласования решений и взаимодействие лиц, принимающих решения (ЛПР). Функции ЛПР выполняют программные агенты. Каждый из них в отдельности, не закрывает всех вопросов, возникающих при автоматизации процессов организационно-технической системы (ОТС). В этой связи анализ существующих динамических моделей процессов ОТС и моделей архитектуры информационных систем, и на их основе, создание метода, совмещающего в

себе эти подходы, а также программного обеспечения для его реализации – системы поддержки принятия решений (СППР) является актуальной работой.

При этом следует отметить следующие моменты. Во-первых, совместное использование структурного и объектно-ориентированного подходов позволяют получить модель информационной системы, понятной как разработчику, так и пользователю – предметному специалисту. Во-вторых, мультиагентный подход и экспертные системы позволяют в комплексе решить вопросы формализации и информатизации процессов принятия решений, которые характерны для ОТС. В-третьих, эффект от информатизации будет намного выше, если решать задачу автоматизации совместно с задачей совершенствования БП. Применение средств имитационного и мультиагентного моделирования на начальных этапах разработки, а затем использование полученные результатов и знаний на этапах автоматизации представляется перспективным направлением развития.

Большой вклад в рассматриваемую тему внесли работы следующих исследователей: Д. В. Александрова, Б. Бозма, Г. Буча, А. М. Вендрова, К. Гейна, С. Л. Гольдштейн, В. И. Городецкого, Г. Н. Калянова, О. В. Карсаева, Б. И. Клебанова, М. Минского, Е. Г. Ойхмана, Э. В. Попова, Дж. Рамбо, У. Ройса, Т. Сарсона, П. О. Скобелева, А. Ю. Филипповича, М. Хаммера, Дж. Чампи, А. Н. Швецова, N. R. Jennings, M. J. Wooldridge, К. А. Аксенова.

Основной целью работы является разработка метода поддержки принятия решений для задачи автоматизации процессов ОТС. Для реализации основной цели исследования в работе решают следующие задачи:

- 1) определить особенности, характерные для ОТС;
- 2) провести анализ существующих методов и средств создания ИС с целью оценки уровня развития данного направления;
- 3) провести анализ существующих моделей формализации процессов ОТС, учитывающих динамику процессов и человеческий фактор, с целью выбора адекватной модели для дальнейшего использования ее при разработке метода;

- 4) разработать концептуальную модель ОТС: выделить и описать основные объекты и взаимосвязи между ними, с целью формализации предметной области;
- 5) разработать концептуальную модель ИС: выделить и описать основные объекты и взаимосвязи между ними, с целью построения алгоритмов разработки информационной системы;
- 6) разработать информационно-алгоритмическое обеспечение, реализующее предложенный метод;
- 7) провести исследование новой программной среды при проведении анализа процессов и разработке ИС.

Объем и структура работы. Монография состоит из введения, четырех глав, заключения, списка литературы. Монография содержит 107 рисунков и 18 таблиц. Список литературы включает 96 наименований.

В первой главе рассмотрены этапы системного анализа при разработке информационных систем. Проведен обзор и сравнительный анализ наиболее распространенных CASE-средств, определены требования к созданию интеллектуальной системы поддержки принятия решений в области разработки информационных систем.

Во второй главе определены требования к методу поддержки принятия решений в области разработки ИС МППР, проведен анализ существующих моделей представления знаний и выбрана фреймово-семантическая модель А. Н. Швецова. Предложены фреймово-семантические модели МППР и ИС, метод преобразования модели МППР в элементы архитектуры ИС.

В третьей главе на основе предложенного метода представлены принципы разработки и технические решения разработанной СППР Bpsim.SD, входящей в состав продуктов семейства Bpsim, приведено описание этой системы.

В четвертой главе представлены примеры использования разработанного продукта в реальных проектах: анализ БП и разработка технического задания (ТЗ) на единую информационную систему (ЕИС) вуза УГТУ-УПИ; для предприятия ЗАО «Ведение реестров компаний» с помощью Bpsim.SD было

разработано несколько дополнительных модулей для системы ведения реестров (СВР) «Вереком» и «Вереком 2».

1. Процесс системного анализа при разработке информационных систем

1.1. Этапы системного анализа при разработке информационных систем

Основная проблема, возникающая при разработке информационной системы (ИС) – это сложность понимания сразу всей системы в целом. Для решения этой проблемы целесообразно применять системный анализ. Такой подход позволяет получить целостное представление об объекте автоматизации и разрабатываемой ИС. Проведение системного анализа объекта автоматизации при разработке информационных систем можно разделить на следующие этапы:

1. Определение и назначение объекта автоматизации.
2. Определение целей разрабатываемой системы.
3. Анализ состояния внутренней и внешней среды автоматизируемого объекта и прогноз их изменений.
4. Построение и анализ моделей автоматизируемых процессов.
5. Разработка новых моделей автоматизируемых процессов с учетом проблем, диагностированных на предыдущем этапе.
6. Разработка модели ИС.
7. Разработка и внедрение полученной системы.

Разработки ИС идет в тесном сотрудничестве пользователей, аналитиков и разработчиков. Первые являются специалистами в предметной области. Вторые получают знания от первых и формулируют требования к разрабатываемой ИС. Разработчики реализуют полученные требования в виде готового продукта.

Далее рассмотрим некоторые этапы подробнее.

1.1.1. Организационно-технические системы

Под объектом автоматизации в работе рассматриваются организационно-технические системы (ОТС), которые представляют собой совокупность

организационной структуры и находящихся в её распоряжении технических средств, т.е. совместно рассматривается человек и информационная система. В работе [1] выделены следующие особенности ОТС: многопараметричность, иерархичность, вероятностное поведение, сложность структуры и алгоритмов поведения.

Процессы, протекающие в ОТС можно разделить на три группы:

- производственные и бизнес-процессы (БП) – процессы, связанные с основной деятельностью предприятия;
- процессы согласования;
- процессы принятия решений.

При автоматизации производственных и БП предприятия аналитики, опрашивая пользователей, получают информацию о структуре и функциях ОТС, строят модель БП, при необходимости, дорабатывают ее. Таким образом, знания пользователя, обработанные аналитиком, преобразуются в техническое задание (ТЗ) на разрабатываемую систему.

Для автоматизации процессов согласования лучше всего подходят мультиагентные системы (МАС).

Процессы принятия решений имеют свои особенности [1]. Прежде всего, данные задачи сложно описать алгоритмически. Решения принимаются по определенным сценариям, для описания которых целесообразно использовать базы знаний и технологии экспертных систем (ЭС). При автоматизации деятельности лица, принимающего решение (ЛПР), возможно использование программных интеллектуальных агентов. Следовательно, для таких процессов ИС должна включать систему поддержки принятия решений (СППР), которая поможет ЛПР, на основе имеющейся информации правильно определить проблему и выбрать оптимальное решение. Следует отметить, что не все алгоритмы и сценарии поведения поддаются полной формализации. В некоторых случаях требуется непосредственное участие ЛПР.

На рисунке 1.1. показаны особенности автоматизации ОТС.

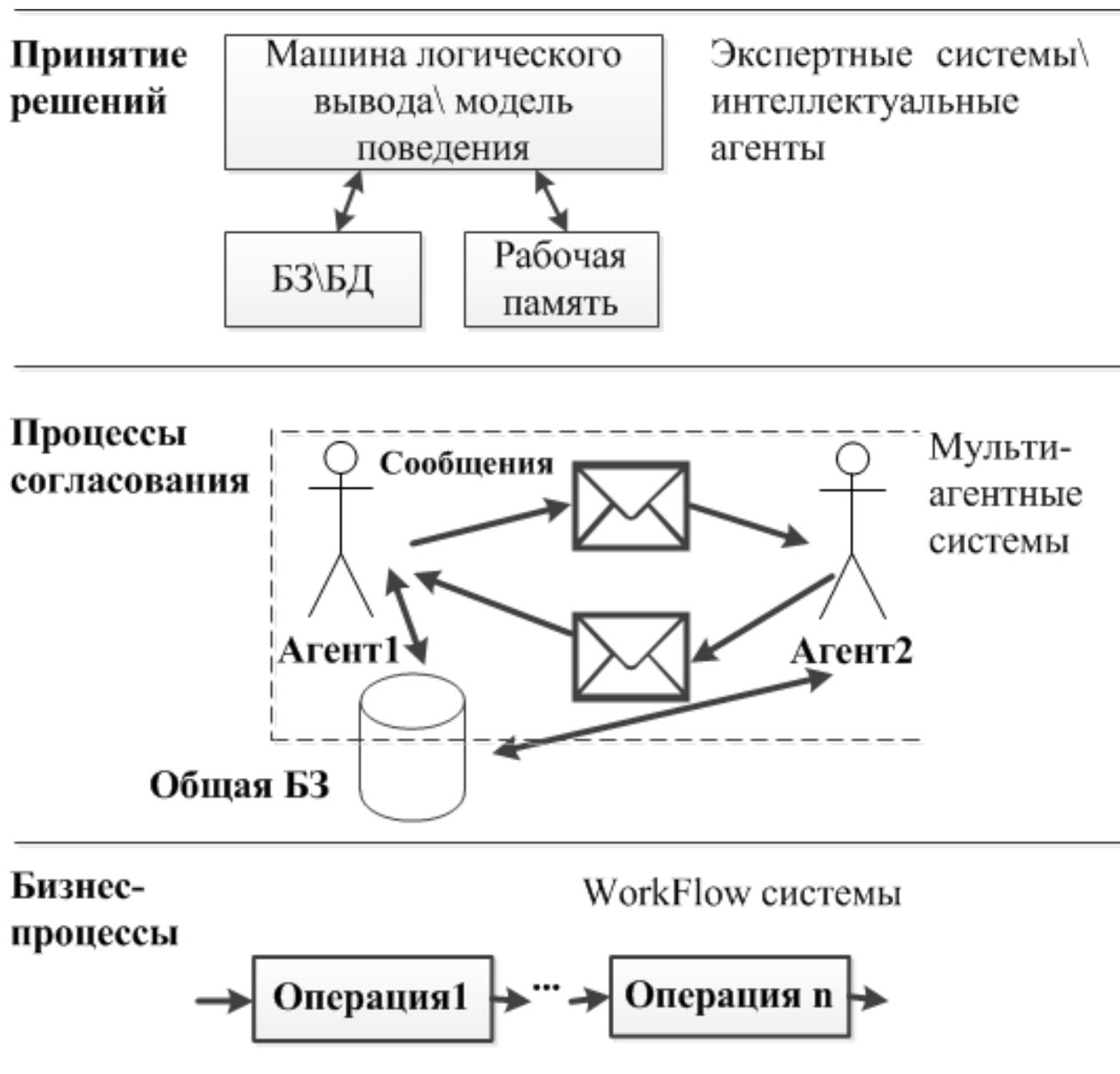


Рисунок 1.1. Особенности автоматизации ОТС

Автоматизация каждой группы процессов, протекающих в ОТС, имеет свои особенности. Существующие подходы к разработке, каждый в отдельности, не закрывает всех вопросов, возникающих при этом. Следовательно, целесообразно совмещать эти методы при разработке информационных систем.

1.1.2. Моделирование автоматизируемых процессов

Замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала с помощью объекта-модели называется моделированием [3]. Для одной и той же системы можно построить

несколько моделей, каждая из которых определяет конкретный аспект системы. При этом используются разные наборы диаграмм и документов.

Успешное решение задачи создания ИС не возможно без четкого представления БП, протекающих на автоматизируемом предприятии. Поэтому, прежде всего, необходимо построить бизнес-модель, которая отражает автоматизируемые процессы и необходимые для них ресурсы.

В процессе анализа предметной области при разработке ИС строятся модели деятельности предприятия. Они могут быть двух видов:

- модели «AS-IS» («как-есть»), которые отражают существующие бизнес-процессы предприятия;
- модели «AS-TO-BE» («как-будет»), которые показывают представления о новых процессах и технологиях работы предприятия.

Переход от модели «AS-IS» к модели «AS-TO-BE» происходит благодаря реинжинирингу (раздел 1.2.3).

Далее, на основе моделей «AS-TO-BE», строятся модели разрабатываемой ИС. Наличие моделей ИС также положительно сказывается на документировании проекта, поскольку принимаемые проектные решения становятся более наглядными. Состав моделей, используемых в каждом конкретном проекте, степень их детализации зависит от следующих факторов [4]:

- сложность разрабатываемой системы;
- необходимой полнотой ее описания;
- знаний и навыков участников проекта;
- времени, отведенного на разработку.

Поскольку проектирование и моделирование тесно связаны друг с другом, то при моделировании БП предприятия и структуры программного обеспечения (ПО) также используют структурные и объектно-ориентированные методы.

Структурные методы анализа

Структурным анализом называется метод исследования системы, начинающий с ее общего обзора, который затем детализируется, приобретая иерархическую структуру со все большим числом уровней [4]. Для него характерно:

- разбиение системы на уровни абстракции с ограничением числа элементов на каждом из уровней;
- ограниченный контекст, включающий лишь существенные на каждом уровне детали;
- использование строгих формальных правил записи;
- последовательное приближение к конечному результату.

В структурном анализе и проектировании используют различные модели, наиболее распространенными являются:

- функциональная модель SADT (IDEF0), которая описывает функциональную структуру системы [4; 5];
- модель IDEF3, которая описывает процессы, в которых важно понять последовательность выполнения действий и взаимозависимости между ними, модели IDEF3 часто используются для детализации функциональных блоков IDEF0 [4; 6];
- диаграммы потоков данных DFD, которые описывают передачу информации между функциональными процессами [4; 7];
- спецификация BPMN (англ. Business Process Model and Notation, нотация и модель бизнес-процессов) позволяет описать БП в виде последовательности объектов потока управления с возможностью задания циклических действий [8].

В семейство IDEF входят и другие модели, которые не так широко используются. Следует отметить методологию онтологического исследования сложных систем IDEF5. Она позволяет описать правила и ограничения, которые представляют состояние системы, с использованием словаря терминов [9].

Можно выделить следующие положительные черты структурных методов:

- графические диаграммы позволяют наглядно представить структуру системы;
- визуальное моделирование доступно неспециалистам в области информационных технологий (ИТ);
- иерархическая структура диаграмм позволяет описывать функции системы с разной степенью подробности.

При разработке ИС целесообразно использовать диаграммы потоков данных DFD [4], поскольку этот метод хорошо согласуется со средством моделирования данных (модель «сущность-связь»). Хранилища данных, описываемые на диаграммах DFD, являются основой для построения модели «сущность-связь».

Объектно-ориентированные методы анализа и проектирования ПО

IDEF4 – методология построения объектно-ориентированных систем. Средства IDEF4 позволяют наглядно отображать структуру объектов и заложенные принципы их взаимодействия, тем самым, позволяя анализировать и оптимизировать сложные объектно-ориентированные системы [10].

Язык UML [11; 12; 13] – широко используемый в настоящее время метод объектно-ориентированного анализа. Язык UML представляет собой набор диаграмм, которые позволяют описать разрабатываемую информационную систему с двух сторон: логическом / физическом и статическом / динамическом. Для описания статических частей системы используются следующие диаграммы: классов, объектов, компонентов и развертывания. Для описания динамической составляющей ИС используются: прецедентов, последовательности, кооперации, состояний и деятельности.

Широкое применения UML во многом связано с возможностью расширения существующих элементов языка, что позволяет создавать языки для использования в конкретной предметной области на основе UML. Этот язык (версия 2.0) включает в себя следующий набор диаграмм [13]:

- 1) вариантов использования – предназначены для определения множества функций, которые будет выполнять ИС;

2) деятельности – позволяет описать деятельность в форме последовательности действий, которые выполняются различными элементами, входящими в состав системы;

3) последовательности – объекты ИС взаимодействуют между собой путем передачи сообщений друг другу; эти диаграммы позволяют описать последовательность передачи сообщений между объектами;

4) коммуникации – предназначены для представления взаимодействия в контексте внутренней архитектуры системы и передаваемых сообщений;

5) обзора взаимодействия – предназначены для представления взаимодействия только в контексте потока управления в некоторой агрегированной форме;

6) временная – представляет изменения состояния отдельных линий жизни или особенностей синхронизации сообщений во времени;

7) конечного автомата – граф, который представляет некоторый конечный автомат;

8) классов – позволяют создавать логическое представление системы, т. е. описать иерархию классов, которые будут реализовывать объекты ИС, методы и свойства этих классов;

9) композитной структуры – позволяет изображать внутреннюю структуру или состав классов и взаимосвязи между ними;

10) пакетов – предназначены для представления размещения элементов модели в пакетах и спецификации зависимостей между пакетами и их элементами;

11) объектов – предназначена для спецификации объектов и связей между ними для фиксированного момента времени;

12) компонентов – предназначена для представления физических компонент системы и их взаимодействия;

13) развертывания – позволяет описать требования к аппаратной части проектируемой ИС: процессоры, устройства и их соединения; строится одна диаграмма для каждой модели.

Моделирование БП является важной составляющей разработки ИС. Существуют структурные и объектно-ориентированные методы анализа и проектирования ПО. Каждый из них имеет свои достоинства и недостатки. Для получения наибольшего эффекта в работе ставится и решается задача интеграции этих методов для задачи разработки ИС.

1.1.3. Реинжиниринг бизнес-процессов

Современное предприятие имеет довольно сложную структуру и алгоритмы работы. На начальном этапе разработки ИС необходимо удостовериться в том, что организация существующих БП позволяет работать предприятию наиболее оптимально, и, если это не так, то внести соответствующие изменения. Для этого используется реинжиниринг бизнес-процессов (РБП) – фундаментальное переосмысление и радикальное перепроектирование БП для достижения коренных улучшений в основных показателях деятельности предприятия [14]. Целью РБП является системная реорганизация всех потоков предприятия (материальных, финансовых и информационных), которая должна привести к упрощению организационной структуры, перераспределению и минимизации использования различных ресурсов, сокращению сроков реализации потребностей клиентов. Все это в конечном итоге позволит повысить качество работы предприятия [15]. Для успешного проведения РБП сначала необходимо определить цели предприятия и критерии эффективности его работы. Затем строится модель существующих БП компании, проводится ее анализ. В результате предлагаются изменения, улучшающие БП предприятия. На этом этапе целесообразно использовать средства имитационного моделирования. Это позволяет оценить эффективность работы на различных этапах БП и определить проблемные места. После этого разрабатывается новая модель БП, и происходит ее внедрение.

Наиболее распространенным средством моделирования динамических процессов (переходов из одного состояния в другое (из одной ситуации в

другую)) является имитационное моделирование и, в частности, дискретно-событийное [16; 17; 18].

Также в имитационном моделировании используется мультиагентный подход. При этом поведение системы определяется как результат деятельности многих агентов. Для построения мультиагентной модели необходимо определить поведение отдельных агентов и взаимоотношения между ними [19].

Для анализа, совершенствования и реинжиниринга БП в ОТС используются средства имитационного и мультиагентного моделирования, что повышает эффективность автоматизации.

Выводы по разделу. В работе рассматриваются этапы разработки информационной системы, предшествующие написанию технического задания. Этот документ является связующим звеном между участниками ИТ-проекта. От качества описания каждой группы процессов, протекающих в ОТС, зависит результат разработки информационной системы. Решение задачи автоматизации совместно с задачей совершенствования БП позволяет повысить эффект от внедрения ИС.

1.2. Риски, связанные с разработкой информационных систем, и пути их снижения

Разработка ИС сопровождается определенными рисками [20]. Перечислим некоторые из них:

1. Сроки выполнения этапов могут отставать от графика проекта или бюджета проекта будет существенно превышен. Это может привести к закрытию проекта до завершения, следовательно, ИС даже не будет разработана.

2. Низкое качество внедренной ИС может привести к отказу от ее использования.

3. Разработанная ИС теряет свою полезность после нескольких лет эксплуатации, поскольку количество недостатков и стоимость внесения

изменений увеличивается настолько, что становится проще и дешевле разработать новую систему, чем поддерживать существующую.

4. В процессе эксплуатации ИС выясняется, что она не решает необходимых задач предприятия. Это может быть следствием изначально неточной постановки задачи или изменений БП в ходе разработки ИС, которые не учитывались.

5. В процессе эксплуатации ИС выясняется, что она имеет множество функций, не используемых заказчиком, а действительно полезные – в системе не реализованы.

6. В течение жизненного цикла ИС происходит почти полное обновление команды разработчиков. Недостаточное документирование системы не позволяет новым сотрудникам успешно модернизировать ИС.

Согласно исследованиям [21], проекты, длящиеся более двух лет, являются довольно рискованными. На рисунке 1.2. показан процент успешных проектов в зависимости от его длительности и характеристик. Успешность проекта определяется разными характеристиками (завершенность в срок, уложились в бюджет и т. п.).

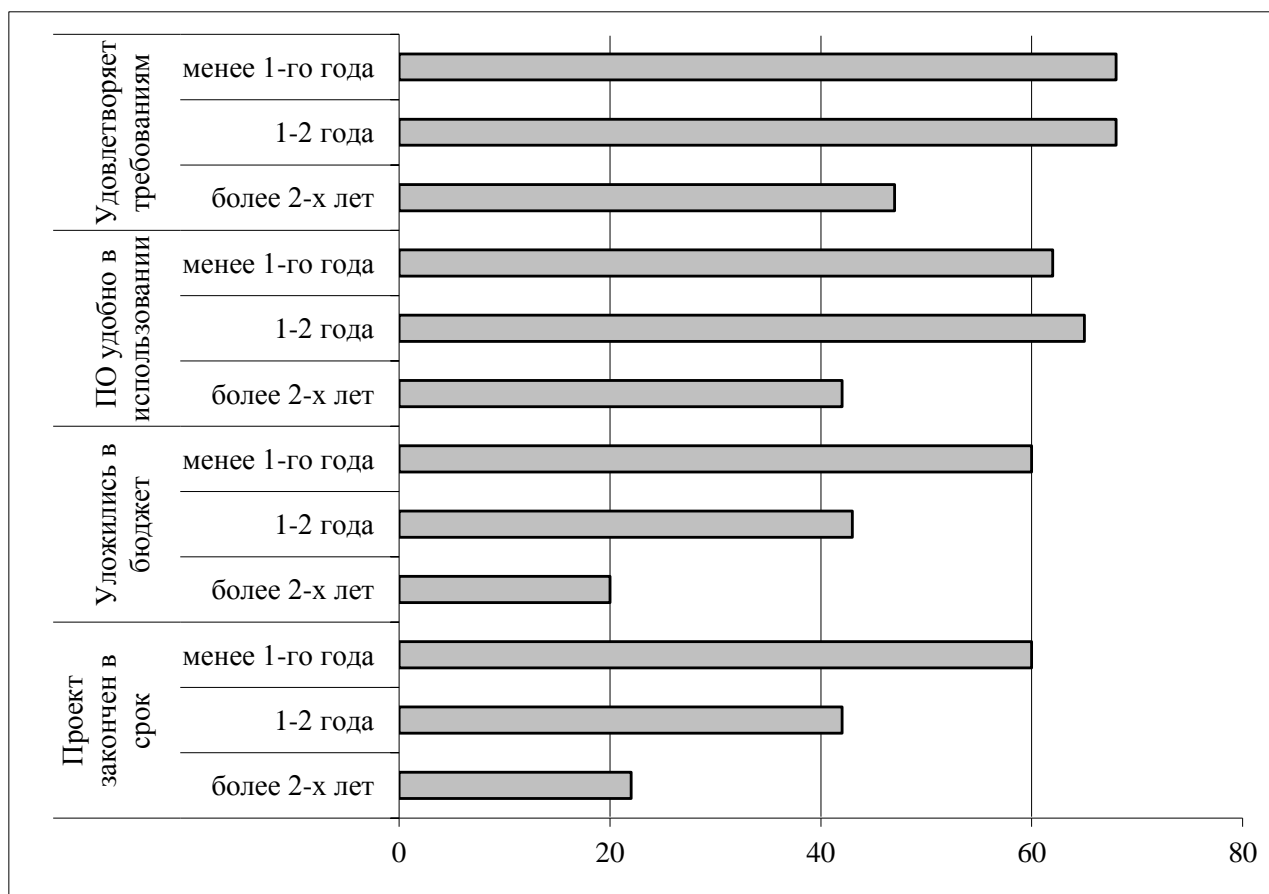


Рисунок 1.2. Процент успешных проектов
в зависимости от его длительности и характеристик

В настоящее время создание ИС рассматривается как формализованный процесс, который должен соответствовать определенным стандартам и нормативным документам [4]. Создаются различные методы и технологии разработки ИС, которые позволяют снизить риски, связанные с ее разработкой, увеличить производительность участников ИТ-проекта, улучшить качество разрабатываемой системы за счет повышения управляемости процесса создания ИС и своевременного учета изменившихся требований к системе.

Следует отметить, что эффективность применения этих методов и технологий напрямую связана с наличием программного средства, обеспечивающего автоматизацию этапов – CASE-средство (Computer Aided Software Engineering – компьютерно-ориентированная программная инженерия).

Метод разработки включает в себя:

- определенную последовательность этапов разработки;
- перечень нотаций (графических и текстовых средств), используемых для описания создаваемой системы;
- критериев и правил, используемых для оценки полученных результатов.

Выводы по разделу: *применение определенных методов разработки ПО позволяет уменьшить возможные риски, а наличие CASE-средств, которые их поддерживают, повышает эффективность использования метода разработки ПО.*

1.3. Методологические и теоретические основы поддержки принятия решений, моделирования и разработки информационных систем

1.3.1. Применение имитационного моделирования при разработке программного обеспечения

Принятие решения о внедрении на предприятии единой ИС связано со многими трудностями. Можно выделить следующие основные проблемы, возникающие при этом:

- 1) наличие нескольких несвязанных ИС, решающих узкие проблемы;
- 2) неоптимальные БП, которые не позволяют эффективно работать предприятию;
- 3) сложность прогнозирования характеристик работы ИС (скорость обработки запросов к базе данных, скорость передачи данных по сети и т.п.).

Для решения первой проблемы необходимо спроектировать новую БД для единой ИС и перенести в нее накопленные данные.

Наиболее эффективным путем решения второй проблемы является построение модели существующих БП и проведение их РБП [3; 22]. Имитационное моделирование (ИМ) на этом этапе существенно помогает в анализе такой сложной системы, как современное предприятие.

Имитационная модель описывает законы функционирования каждого элемента системы и их взаимосвязи. В результате имитационного эксперимента по исходным данным можно получить сведения о состояниях БП в определенные моменты времени. Это позволяет оценить характеристики системы и лучше понять, как разработать систему, удовлетворяющую заданным критериям оценки эффективности [3]. Для получения достоверных и точных результатов ИМ при небольших затратах машинного времени необходимо осуществлять планирование экспериментов. Оно бывает стратегическим и тактическим [3]. В первом случае, определяют какие факторы и какие их уровни использовать в эксперименте, чтобы получить максимум информации о системе без полного перебора. Во втором случае, определяют способы проведения экспериментов (длительность прогона, точность результата).

Для прогнозирования характеристик работы ИС можно расширить имитационную модель так, чтобы она включала в себя не только описания автоматизируемых БП, их статистические характеристики, такие как время обработки заявки (документа), время простоя в очереди, количественные характеристики процессов, а также временные и количественные характеристики транзакций в ИС. Эксперименты с такой имитационной моделью позволят до реализации готовой ИС получить представление о ее производительности, требования к скорости и интенсивности обработки заявок (документов) и транзакций в БД.

Применение ИМ при разработке ИС целесообразно как на этапе анализа автоматизируемых БП с целью их оптимизации, так и перед разработкой ИС для прогнозирования характеристик ее работы.

1.3.2. Использование систем искусственного интеллекта при разработке информационных систем

Использование искусственного интеллекта (ИИ) при разработке ИС переводит этот процесс на качественно новый уровень, поскольку позволяет автоматизировать не только рутинные операции, но и процесс принятия решений при создании ИС. Для этого инструментарий, поддерживающий процесс разработки, должен включать в себя ЭС. Поскольку невозможно полностью исключить человека из процесса разработки ИС, то целесообразно использовать диалоговые ЭС [23].

В БЗ интеллектуальной СППР в области разработки ПО (CASE-средстве) должны храниться факты и правила, полученные от экспертов предметной области, а также профессиональные знания разработчиков. Использование такой СППР позволят решать практические задачи, возникающие в процессе разработки ИС.

Развитие информационных технологий определило повышенное внимание к интегрированному использованию ИС и ЭС в части автоматизации деятельности ЛПР. Такие системы могут быть использованы в задачах, для решения которых требуется взаимодействие целого ряда экспертов предметной

области. Использование БЗ и машины логического вывода (МЛВ) в ИС позволят автоматизировать поиск решения, к которому мог бы прийти в аналогичной ситуации эксперт [23].

Учет большого объема данных, знаний экспертов-специалистов, формирование ответа на основании некоторого аналога процесса рассуждений, диалог с пользователем – все это делает ЭС тем инструментом, который позволит сделать более интеллектуальным процесс создания ИС и автоматизировать работу ЛПР.

1.3.3. Применение мультиагентного подхода

Мультиагентные системы (МАС) представляют собой новое направление развития искусственного интеллекта, которое сформировалось на основе результатов исследований в области распределенных компьютерных систем, сетевых технологий решения проблем и параллельных вычислений [24].

Агентно-ориентированный подход уже используется при распределенном решении сложных задач, реинжиниринге предприятий, электронном бизнесе, логистике, а также моделировании. При классификации Д. А. Поспелов [25] выделяет два класса задач:

- 1) задачи распределенного управления и задачи планирования достижения целей, в которых агенты совместно решают поставленные проблемы, и необходимо обеспечение эффективного способа кооперации их деятельности;
- 2) локальные задачи, при решении которых агенты используют общие, как правило, ограниченные ресурсы.

Агент – это аппаратная или программная сущность, способная действовать в интересах достижения целей, поставленных перед ней владельцем и/или пользователем [26]. Агенты описываются так же рядом свойств, которые характеризуют понятие агента. Обычно агент обладает набором из следующих свойств [24]:

- адаптивность: агент обладает способностью обучаться;

- автономность: агент работает как самостоятельная программа, ставя себе цели и выполняя действия для достижения этих целей;
- сотрудничество: агент может взаимодействовать с другими агентами несколькими способами, например, играя роль поставщика / потребителя информации или одновременно обе эти роли;
- способность к рассуждениям: агенты могут обладать частичными знаниями или механизмами вывода, например, знаниями, как приводить данные из различных источников к одному виду. Агенты могут специализироваться на конкретной предметной области.
- коммуникативность: агенты могут общаться с другими агентами;
- мобильность: способность к передачи кода агента с одного сервера на другой.

Следует отметить, что агенты могут быть отнесены к следующим типам:

- реактивные;
- интеллектуальные;
- гибридные.

Реактивный агент принимают решения на основе знаний «ситуация-действие». Интеллектуальный агент решает поставленные перед ним задачи, исходя из своих целей, используя общие ограниченные ресурсы и знания о внешнем мире. Гибридный агент сочетает в себе возможности первых двух.

Интеллектуальная МАС представляет собой множество интеллектуальных агентов, распределенных в сети. Они отслеживают необходимые данные и взаимодействуют друг с другом для достижения поставленных перед ними целей. Можно выделить несколько важных причин взаимодействия агентов: совместимость целей (общая цель); отношение к ресурсам; необходимость привлечения недостающего опыта; взаимные обязательства [24].

С точки зрения программирования, агенты представляют собой ПО, которое способно действовать самостоятельно от лица пользователя. При создании МАС может использоваться архитектура «клиент-сервер». В

зависимости от поставленной задачи, возможны две ее реализации. Первая, «толстый клиент – тонкий сервер» – серверная часть реализует доступ к ресурсам, а приложения находятся на компьютерах клиентов. Вторая, «тонкий клиент – толстый сервер» – в клиентской части реализован только интерфейс, а сервер выполняет все остальные задачи ПО.

Введение программных агентов в ИС позволит, в некоторой степени, упростить работу пользователей, поскольку агенты смогут отслеживать состояния системы и предлагать определенные решения.

Важной проблемой при разработке ИС является получение знаний экспертов в определенной предметной области. Эти знания большей частью не формализованы, а поэтому недоступны другим людям. Удачное решение этой проблемы – добавление в создаваемую ИС общей БЗ и разработка агентов, которые на основе этих знаний, будут предлагать решения определенных проблем в автоматизируемой области. Таким образом, проектируемая ИС расширяется интеллектуальной СППР, которая выполняет определенные формализованные функции пользователей и оказывает поддержку при решении задач организационно-технического управления.

Также мультиагентный подход может быть применен при разработке распределенных ИС. В этом случае каждый узел системы представляет собой программного агента со своими ресурсами. Их кооперация обеспечивает функционирование системы в целом.

Из данного раздела вытекают следующие требования к методу разработки ИС:

- возможность разработки модели лица, принимающего решения (ЛПР), представленного интеллектуальным агентом (ИА);*
- возможность описания сценариев принятия решения ИА.*

1.4. Обзор и сравнительный анализ существующих систем поддержки принятия решений в области разработки информационных систем (CASE-средств)

1.4.1. Классификация CASE-средств

Современные CASE-средства охватывают обширную область поддержки многочисленных технологий проектирования ИС: от простых средств анализа и документирования до полномасштабных средств автоматизации, покрывающих весь жизненный цикл ПО [27].

Этапы анализа и проектирования являются наиболее трудоемкими при разработке ИС, поэтому использование CASE-средств позволяет повысить качество принимаемых технических решений и облегчить подготовку проектной документации. CASE-средства позволяют строить различные виды диаграмм, при этом проверяя существующие для них синтаксические правила. Использование графических средств моделирования предметной области позволяет более наглядно представить информацию для аналитиков и разработчиков и оценить последствия изменений, вносимых в БП, для разрабатываемой ИС.

Современный рынок программных средств насчитывает сотни различных CASE-средств [27]. Среди них есть достаточно дешевые системы для персональных компьютеров с весьма ограниченными возможностями, и мощные дорогостоящие системы, поддерживающие весь жизненный цикл ПО.

Обычно к CASE-средствам относят любое программное средство, автоматизирующее ту или иную совокупность процессов жизненного цикла ПО и обладающее следующими основными характерными особенностями [27]:

- мощные графические средства для описания и документирования ИС, обеспечивающие удобный интерфейс с разработчиком и развивающие его творческие возможности;
- интеграция отдельных компонент CASE-средств, обеспечивающая управляемость процессом разработки ИС;

– использование специальным образом организованного хранилища проектных метаданных.

Наиболее используемыми CASE-средствами являются CA Erwin Modeling Suite, Rational Suite, ARIS ToolSet, Power Designer и Borland Together Designer, BizAgi BPM Suit, ELMA BPM, в следующем подразделе приводится их описание.

1.4.2. Описание CASE-средств

CA Erwin Modeling Suite

CA Erwin Modeling Suite – комплект инструментов фирмы CA Technologies, которые в полной мере обеспечивают решение всех задач анализа, проектирования, генерации, тестирования и сопровождения информационных систем [28; 29]. Кратко опишем продукты, входящие в этот пакет.

Process Modeler (ранее Bpwin) [30; 31] – инструмент визуального моделирования бизнес-процессов. Дает возможность представить любую деятельность или структуру в виде модели. Визуальное моделирование позволяет оптимизировать работу организации, проверить ее на соответствие стандартам ISO 9000, спроектировать организационную структуру, исключить ненужные операции. Process Modeler поддерживает три нотации моделирования: IDEF0, IDEF3 и DFD.

Erwin Data Modeler (Erwin) позволяет проектировать, документировать и сопровождать БД и хранилища данных. Визуальная модель базы данных облегчает оптимизацию структуры БД, позволяя добиться её полного соответствия требованиям и задачам предприятия. Erwin поддерживает следующие СУБД: СУБД DB2 для LUW и СУБД DB2 для z/OS, СУБД SQL Server 2008, и поддержка СУБД Teradata 13.0, Oracle, InterBase, Ingres.

Component Modeler (Paradigm Plus) [32] – средство для моделирования компонентов ПО и генерации объектного кода приложений на основе созданных моделей. Продукт можно использовать как при создании новых приложений, так и при изменении или объединении существующих.

Интегрирован с Process Modeler, что дает дополнительные возможности при работе с функциональными моделями. Component Modeler обеспечивает полную поддержку UML, поддерживает синхронизацию проектирования и реализации приложения. Component Modeler поддерживает кодогенерацию и реверс-инжиниринг программных сборок Microsoft .NET (C# и Visual Basic), Microsoft Visual J++ и Microsoft Visual C++.

Таким образом, CA Erwin Modeling Suite позволяет описывать модель БП в нотациях IDEF0, IDEF3, DFD, разрабатывать модель данных и архитектуру ПО с поддержкой различных СУБД. Для анализа и оптимизации БП предлагается использовать функционально-стоимостной анализ. В настоящее время данный пакет особое внимание уделяет моделированию.

Продукты компании IBM Rational

Продукты компании IBM Rational [33; 34; 35; 36] представляет собой комплекс инструментальных средств, поддерживающих весь жизненный цикл разработки ПО и интегрированный с IBM Rational Unified Process (RUP). Технически RUP оформлена в виде размещенной на Web базы знаний, которая снабжена поисковой системой. Она включает в себя: обширные руководства для разработчиков для каждого временного интервала жизненного цикла ПО; руководства по использованию инструментальных средств; примеры и шаблоны.

CASE-средство IBM Rational Software Architect предоставляет разработчику следующие возможности: анализ и проектирование ПО с помощью UML, анализ и контроль структуры приложений Java, подготовка проектной документации. Его компоненты основаны на технологии Eclipse. Расширение Simulation позволяет провести имитационное моделирование поведения ПО на основе UML-диаграмм деятельности, последовательности, коммуникации или диаграммы состояний. Rational Software Architect поддерживает кодогенерацию и реверс-инжиниринг: Microsoft Visual Studio 2010 и .NET Framework 4 для C# и VB.NET, Java и C++.

Технология создания ПО, реализованная в IBM Rational, также может применяться для проектирования пользовательского интерфейса (ПИ). Разработка ПИ основана на концепции сценариев вариантов использования (use-case storyboard), и представляет собой модель взаимодействия пользователя с ПИ. Такой подход позволяет качественно оценить предложенные решения прежде, чем реальный интерфейс будет спроектирован и реализован. Рассмотрим последовательность проектирования ПИ разрабатываемой системы:

1. Проектируются диаграммы вариантов использования, демонстрирующие взаимодействие пользователя с ПИ.

2. Для каждого варианта использования строится диаграмма последовательности, в которой показаны все возможные взаимодействия пользователя с элементами ПИ.

3. Каждая форма приложения представляется на диаграмме классов отдельным классом, элементы формы – атрибуты класса, действия пользователя и системы, связанные с окном – методы класса. Для описания сложных форм используется наследование классов.

На основе полученных моделей ПИ строятся прототипы.

Отметим еще несколько продуктов: IBM Rational Requirements Composer – занимается определением и управлением требованиями на всех этапах жизненного цикла ПО; IBM Rational Team Concert – инструмент для управления конфигурациями ПО, сборок и формирования отчетности; Rational Quality Manager предоставляет возможности для планирования, выполнения и отслеживания состояния ручного и автоматизированного тестирования ПО.

IBM Rational Rose Data Modeler представляет собой средство моделирования данных в нотации ER

Продукты компании IBM Rational уделяют внимание разработке архитектуры организации и программного обеспечения, управлению жизненным циклом созданной информационной системы, не затрагивая вопросов, связанных с моделированием данных.

ARIS ToolSet

Интегрированная среда ARIS ToolSet [37] представляет собой комплекс инструментов для:

- проектирования и управления предприятием;
- моделирования, анализа и оценки бизнес-процессов;
- документирования бизнес-процессов в соответствии с требованиями международных стандартов;
- разработки, внедрения и сопровождения ИС.

Методология моделирования и проектирования ИС, реализованная в этой системе, основывается на совокупности различных методов моделирования, отражающих разные взгляды на проектируемую систему (организация, функции и цели, данные, продукты и услуги, процессы). При построении моделей в ARIS ToolSet можно использовать как собственные методы моделирования ARIS, так и известные языки моделирования, например, UML, BPMN.

В состав ARIS ToolSet входят следующие дополнительные модули:

- ARIS ABC – для функционально-стоимостного анализа бизнес-процессов;
- ARIS Simulation – для имитационного моделирования различных БП;
- ARIS Weblink – для организации удаленной работы;
- ARIS Connectivity for Lotus Notes – работа с документами на базе Lotus Notes;
- ARIS Connectivity for SAP/R3 HR, DataModels – для организации взаимодействия с соответствующими модулями системы R/3;
- различные интерфейсы со средствами разработки (Designer/2000, ERWin, PowerDesigner, Rational Software Architect и др.) – для реализации последующих этапов проектирования ИС, анализа спецификаций и внедрения.
- ARIS составляет основу пакета Business Process Analysis Suite корпорации Oracle, что позволяет автоматически преобразовывать модели процессов в компоненты SOA (англ. service-oriented architecture, сервис-ориентированная архитектура) [38].

Интегрированная среда ARIS ToolSet занимается имитационным моделированием БП, их анализом, совершенствованием и оптимизацией, есть возможность использовать нотацию ER для разработки модели данных. Вопросы, касающиеся разработки архитектуры ПО решаются с помощью других приложений, с которыми у ARIS есть интеграция.

Power Designer

Power Designer представляет собой CASE-средство фирмы Sybase и имеет модульную архитектуру [39]. Power Designer имеет следующие возможности:

- структурное моделирование бизнес-процессов;
- концептуальное и физическое проектирование и генерация БД (поддерживает более 60 СУБД);
- объектно-ориентированный анализ и моделирование данных с использованием UML;
- Интеграция с ведущими средами разработки (Eclipse, Microsoft Visual Studio®, PowerBuilder).

Power Designer поддерживает моделирование данных, статическое моделирование БП и приложений. Напрямую не затрагивает следующие вопросы: управление требованиями и конфигурациями ПО; имитационное моделирование.

Borland Together 2008

Borland Together [13; 40] представляет собой пакет программ фирмы Micro Focus/Borland. Он предназначен для создания моделей, представляющих собой схему бизнес-процессов, структуру данных и архитектуру приложений и предприятия. Borland Together состоит из следующих программ: Borland Together Designer, Borland Together Designer Community Edition и Borland Together Developer.

Borland Together предоставляет следующие возможности:

- визуальное моделирование метамоделей для определенной предметной области;

- моделирование бизнес-процессов с использованием нотаций Business Process Modeling, BPMN;
- разработка проектов графических моделей программных приложений в нотации языка UML;
- разработка логических диаграмм данных «сущность-связь» в нотации ER и IDEF1x;
- прямое и обратное проектирование для ведущих СУБД (Oracle, DB2, Sybase, MS SQLServer);
- генерирование документации;
- генерирование программного кода для Java, C++ и C#;
- распознавание шаблонов проекта исходного кода;
- генерация схем контрольных примеров из требований, а также трассирование элементов моделей от требований и к требованиям при помощи программ CaliberRM и RequisitePro.

Borland Together 2008 является CASE-средством, которое автоматизирует процесс разработки ПО на этапах моделирования БП, создания диаграмм «сущность-связь» с возможностью генерации структуры БП, моделирования ПО, заканчивающееся генерированием программного кода.

BizAgi BPM Suit

Система BizAgi BPM Suit [41; 42] представляет собой платформу для автоматизации БП. Для моделирования БП используется приложение BizAgi Process Modeler, которое использует нотацию BPMN. Bizagi Studio позволяет преобразовать описанный БП в работающее приложение без участия программиста. Модель находится в хранилище сервера, интерпретируется и выполняется через Web-приложение на Bizagi BPM Server. Этот сервер позволяет анализировать различные показатели выполнения БП с целью выявления проблемных мест и их улучшения. Есть доступ к внешним источникам данных (Oracle, MS SQL Server).

Важно сделать замечания относительно функционала приложений, разрабатываемых с помощью BizAgi, и моделируемых процессов: 1)

функциональность приложений относится к классу Work Flow; 2) моделирование касается только организационных процессов.

Система BizAgi BPM Suit относится к классу систем управления БП и позволяет разрабатывать проблемно-ориентированные приложения класса Work Flow, с возможностями аналитики и контроля. Данная система не поддерживает проектирование архитектуры более широкого класса ПО.

ELMA BPM Suite

ELMA BPM Suite [43] включает в себя приложение для управления показателями БП и управление БП. В программе «Дизайнер ELMA» проводится моделирование БП с использованием нотации BPMN. Спроектированная модель хранится на сервере приложения. Далее происходит выполнение БП, причем можно запускать на исполнение несколько экземпляров одного и того же БП. Пользователь получает информацию о задачах, которые ему необходимо выполнить, в виде страниц в браузере. Система автоматически формирует задачи для каждого сотрудника при выполнении БП. С помощью специальных приложений осуществляется мониторинг и контроль БП.

Система управления БП ELMA BPM Suite позволяет разработать внутренний портал предприятия для управления его работой. Кроме того, предлагаются готовые решения для электронного документооборота, управления проектами. ELMA не поддерживает UML и проектирование широкого класса ПО, имитационное моделирование процессов.

1.4.3. Критерии сравнения функциональных возможностей CASE-средств

Для сравнительной оценки функциональных возможностей CASE-средств предлагается следующий набор критериев:

1. Нотации, используемые при описании бизнес-процессов и архитектуры ИС (IDEF0, DFD, UML, BPMN).
2. Использование имитационного моделирования для анализа, совершенствования и оптимизации БП.

3. Проектирование ПО на основе имитационной модели БП. Данная возможность позволяет использовать информацию из модели при проектировании ПО, что снижает трудоемкость работы.

4. Разработка моделей лиц, принимающих решения (ЛПР).

5. Описания сценариев принятия решений ЛПР.

6. Проектирование пользовательского интерфейса. Разработка макета (предварительного прототипа) пользовательского интерфейса.

7. Конвертация одних диаграмм в другие. При этом для построения новых диаграмм используется информация, уже отраженная в первых диаграммах, а не строить диаграммы с начала.

8. Проектирования структуры БД.

9. Возможность генерации исполняемого кода.

10. Генерация технической документации к проектируемой ИС. Поскольку информация по проекту хранится в единой БД и автоматически обновляется при внесении в него изменений, то в любой момент имеется возможность автоматически создавать актуальную техническую документацию.

1.4.4. Сравнительный анализ CASE-средств

Ниже приводится сравнение программных продуктов ARIS ToolSet, Power Designer, Borland Together Designer, Продукты IBM Rational, CA Erwin Modeling Suite, Biz Agi и Elma в области проектирования ПО (таблица 1.1). Они являются наиболее распространенными среди пользователей – разработчиков программных систем.

Таблица 1.1

Сравнение CASE-средств

Критерии сравнения	Продукты IBM Rational	CA Erwin Modeling Suite	ARIS ToolSet	Power Designer	Borland Together Designer	Biz Agi	Elma
Поддержка IDEF0, DFD	НЕТ	ДА	ДА	НЕТ	НЕТ	НЕТ	НЕТ
Поддержка UML	ДА	ДА	ДА	ДА	ДА	НЕТ	НЕТ
Поддержка BPMN	НЕТ	НЕТ	ДА	НЕТ	ДА	ДА	ДА

Критерии сравнения	Продукты IBM Rational	CA Erwin Modeling Suite	ARIS ToolSet	Power Designer	Borland Together Designer	Biz Agi	Elma
ИМ БП	НЕТ	НЕТ	ДА	НЕТ	НЕТ	ДА	НЕТ
Проектирование ПО на основе ИМ БП	НЕТ	НЕТ	НЕТ	НЕТ	НЕТ	ДА	ДА
Разработка модели ЛППР	НЕТ	НЕТ	НЕТ	НЕТ	НЕТ	ДА	ДА
Описание сценария принятия решений ЛППР	НЕТ	НЕТ	НЕТ	НЕТ	НЕТ	ДА	НЕТ
Проектирование ПИ	НЕТ	НЕТ	НЕТ	НЕТ	НЕТ	ДА	НЕТ
Конвертация диаграмм	НЕТ	НЕТ	НЕТ	НЕТ	НЕТ	НЕТ	НЕТ
Проектирование структуры БД	ДА	ДА	ДА	ДА	ДА	НЕТ	НЕТ
Генерация тех. документации к создаваемой ИС	ДА	ДА	ДА	ДА	ДА	ДА	НЕТ

В пакете CA Erwin Modeling Suite отсутствуют средства ИМ БП, но есть возможность экспортировать модель в систему моделирования Arena.

Aris Toolset, помимо указанных нотаций, использует большое количество других нотаций (диаграммы Чена, Object Modeling Technique). Следует отметить, что большое количество нотаций требует дополнительных знаний от аналитика для их использования. В пакете Aris Toolset имеется адаптированный движок моделирования, к недостаткам которого можно отнести отсутствие поддержки моделирования систем массового обслуживания [44].

Power Designer и Borland Together Designer прежде всего являются средствами для разработки UML-диаграмм и генерирование программного кода, они не предоставляют возможностей для анализа и имитационного моделирования процессов.

Рассмотренные системы класса BPM подходят для разработки узкого класса систем.

Все рассмотренные продукты условно можно разделить на четыре группы:

1. Системы, охватывающие весь жизненный цикл ПО (продукты IBM Rational, CA Erwin Modeling Suite).

2. Системы, автоматизирующие разработку ПО (Power Designer и Borland Together Designer).

3. Системы анализа и имитационного моделирования БП (Aris Toolset).

4. Системы управления БП (BizAgi и Elma).

Поскольку в работе решаются задачи автоматизации процессов ОТС, то наилучшими вариантами представляются системы группы 3 (Aris Toolset и Bpsim).

К недостаткам рассматриваемых CASE-средств можно отнести следующее:

– отсутствует интеграции структурного и объектно-ориентированного подхода;

– отсутствие интеллектуальности процесса проектирования – не решена задача автоматического перехода к проектированию одних диаграмм на основе других;

– управление БП только определенного класса систем (Work Flow);

– отсутствие пакетов для ИМ БП, а следовательно, и возможность использовать информацию из модели при проектировании ИС.

Предлагается разработать метод поддержки принятия решений при разработке ИС и CASE-средство, которые бы устранили эти недостатки.

2. Метод поддержки принятия решений при разработке информационных систем для предметной области мультиагентных процессов преобразования ресурсов

2.1. Требования к модели и методу поддержки принятия решений при разработке информационных систем

Будем рассматривать разработку ИС предметной области мультиагентных процессов преобразования ресурсов (МППР). На основе проведенного анализа методологических и теоретических основ поддержки принятия решений, моделирования и разработки ИС (см. раздел 1.3) и CASE-средств (см. раздел 1.4.4) сформулируем требования к методу поддержки принятия решений (ППР) в области разработки ИС.

Метод ППР при разработке ИС должен обеспечивать автоматизацию процесса создания ИС для предметной области организационно-технических систем. Таким образом, можно выделить следующие требования:

- 1) выбор методики системного анализа и модели для формализации процессов ОТС. При этом будем учитывать наличие ЛПР, которые могут быть представлены в виде ИА;
- 2) ИМ для проверки модели «как будет» на этапе реинжиниринга БП, и оценки производительности ИС.
- 3) интеллектуальная разработка ИС, включающая функциональный и объектно-ориентированный анализ, моделирование ПИ, формирование исполняемого кода ИС.

Рассмотрим принципы построения интеллектуальных СППР в области разработки ПО (CASE-средств). Структура такой системы включает в себя диалоговую ЭС, состоящую из базы знаний (БЗ), БД, механизма логического вывода (МЛВ), блок объяснения полученных решения, блок обучения (адаптация ЭС к изменяющейся действительности), блок понимания, блок ведения, пополнения и корректировки БЗ.

В разделе 2.2 проведем сравнение различных моделей БП и выберем одну для представления ППР. Задача выбора модели представления знаний о предметной области решается в разделе 2.3.

2.2. Выбор модели представления бизнес-процессов

В работе [44] был проведен анализ возможностей использования наиболее распространенных математических моделей дискретных процессов для представления ППР. Было показано, что модели: сети Петри; расширенные сети Петри; системы массового обслуживания (СМО); модели системной динамики не обеспечивают всех требований для моделирования ППР. Математическая модель ППР, представленная в работе [44], далее была расширена аппаратом мультиагентных систем [46].

Динамическая модель МППР была получена К. А. Аксеновым и Н. В. Гончаровой [47]. Она состоит: процессы (PR), операции (Op), ресурсы (RES), команды управления (U), средства ($MECH$), источники ($Sender$) и приемники ресурсов ($Receiver$), перекрестки ($Junction$), параметры (P), агенты ($Agent$). Отдельно выделены информационные типы ресурсов: сообщения ($Message$) и заявки на выполнение операции ($Order$). Описание причинно-следственных связей между элементами преобразования и ресурсами задается объектом «связь» ($Relation$).

i -ая операция (Op_i) представлена следующей структурой:

$$Op_i = \langle f, RESin_i, RESout_i, MECH_i \rangle, \quad (2.1)$$

где f – функция, которую реализует i -ой операция;

$RESin_i$ – входные ресурсы для выполнения i -ой операции;

$RESout_i$ – выходные ресурсы для выполнения i -ой операции, $RESout_i = f(RESin_i)$;

$MECH_i$ – механизмы для выполнения i -ой операции;

$RESin_i = \{RESin_{i1}, RESin_{i2}, \dots, RESin_{ik}\}$ – множество входных ресурсов

$RESout_i = \{RESout_{i1}, RESout_{i2}, \dots, RESout_{im}\}$ – множество выходных ресурсов

$Sender_{ik}$ – источник ik -ресурса;

$Receiver_{ik}$ – приемники ik -ресурса.

Семантика процесса преобразования ресурса показана на рисунке 2.1.

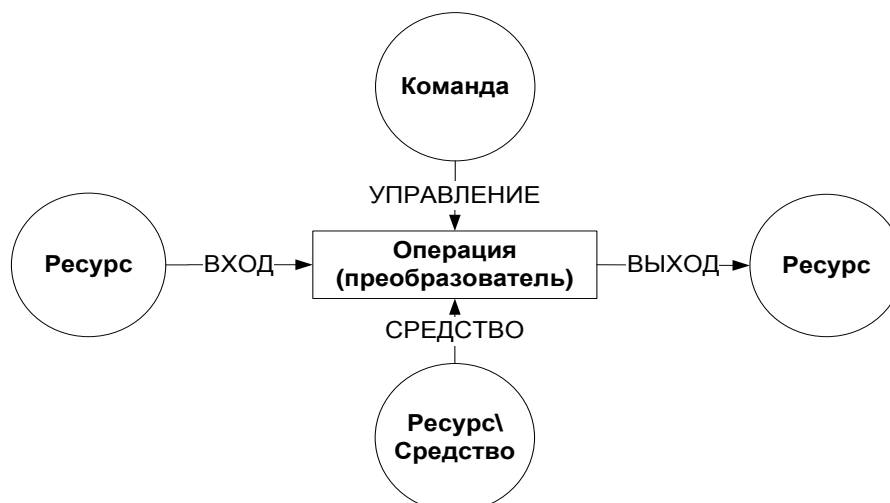


Рисунок 2.1. Семантика процесса преобразования ресурса

Результаты сравнения приведем в виде таблицы 2.1.

Таблица 2.1

Результаты сравнения математических моделей дискретных процессов

Характеристики	сети Петри	расширенные сети Петри	СМО	модели системной динамики	модель МППР
Учет временных характеристик	НЕТ	ДА	ДА	ДА	ДА
Возможность учета различных типов ресурсов	НЕТ	ДА	ДА	НЕТ	ДА
Моделирование конфликтов на общих средствах	НЕТ	НЕТ	ДА	НЕТ	ДА
Модель ЛПР (ИА)	НЕТ	НЕТ	НЕТ	НЕТ	ДА

Полученную математическую модель МППР предлагается использовать в качестве модели описания БП. К преимуществу модели МППР относится возможность использования моделей ЛПР.

Одной из задач при разработке CASE-средств является задача выбора модели представления знаний (для реализации семантики перехода от объектов автоматизируемого процесса к объектам предметной области ИС). Эта задача решается в разделе 2.4.

2.3. Выбор модели представления знаний

Рассмотрим существующие способы представления знаний [48; 49]. При анализе будет учитывать следующие требования.

1. Наиболее простой и естественный переход от неформализованных знаний и представлений к формальным моделям для описания БП и ИС, наглядность представления информации для пользователя.

2. Удобство представления иерархических данных, поскольку предметные области МППР и ИС образуют иерархию.

3. Простота добавления новых знаний.

4. Техническая реализация выбранной модели должна быть достаточно простой и согласовываться с объектно-ориентированным подходом (ООП) разработки ПО.

5. Возможность использования языка UML в качестве визуального языка представления знаний (объектно-ориентированный анализ).

1 способ – Представление знаний в виде продукций или правил и их интерпретатор, который определяет когда и какое правило применяется.

Продукционная модель определяется следующим образом:

$$P_1, \dots, P_m \rightarrow Q_1, \dots, Q_n, \quad (2.2)$$

где P_1, \dots, P_m – предпосылки,

Q_1, \dots, Q_n – действия, выполняемые в случае истинности предпосылок.

Принцип работы продукционной системы заключается в следующем [23]: выполняется продукция (правило), условие которой окажется истинным для текущего состояния БЗ и БД. При этом, это правило активирует данные, находящиеся в заданной структуре БД. Правила выполняются до тех пор, пока они все не выполнятся или не вступит в действие правило остановки.

В продукционную модель достаточно просто добавлять новые знания, поскольку любая продукция может размещаться в любом месте модели. Механизм вывода в продукциях хорошо сочетается с процедурным подходом в программировании.

В качестве недостатков можно указать следующее: неудобно реализовывать иерархическую структуру, ненаглядное представление знаний и неэффективный процесс вывода, поскольку, в общем случае необходимо проверить применимость всех правил.

2 способ – Представление знаний в виде семантической сети. *Семантическая сеть* – это ориентированная графовая структура, каждая вершина которой отображает некоторое понятие (объект, процесс, ситуацию), а ребра графа соответствуют отношениям типа «это есть», «принадлежать», «быть причиной», «входить в», «состоять из», «быть как» и аналогичным между парами понятий [49].

Семантическая модель определяется следующим образом

$$S=(O,R_1,R_2,...,R_n) \quad (2.3)$$

где O – множество объектов ИС;

$R_i \ /i=1,n$ – множество отношений между объектами;

I – тип отношений.

Можно указать следующие достоинства данного способа: наглядность представления знаний для пользователей, семантическая модель хорошо сочетается с иерархическими знаниями. К недостаткам можно отнести сложность реализации механизма вывода.

3 способ – Представление знаний в виде фреймов. Минский в своей работе [50] определил фрейм как «структуру данных для представления стереотипных ситуаций». Фреймовая модель представления знаний задает остов описания класса объектов и удобна для описания структуры и характеристик однотипных объектов (процессов, событий) описываемых фреймами – специальными ячейками (шаблонами понятий) фреймовой сети (знания) [49].

Фрейм состоит из слотов, которые представляют собой различные характеристики объекта (атрибуты), и наполнителей (значения этих атрибутов и процедуры, выполняющиеся при изменении данных фрейма). У каждого

фрейма есть специальный слот, который хранит наименование сущности, которую он представляет. Модель фрейма можно представить следующим образом:

$$\Phi = \langle ИФ, ТФ, ССЛ \rangle, \quad (2.4)$$

$$ССЛ = \{Сатр\}, \quad (2.5)$$

$$Сатр = \{(Иатр_1, МО_1, ЗА_1), (Иатр_2, МО_2, ЗА_2), ..., (Иатр_m, МО_m, ЗА_m)\}. \quad (2.6)$$

где *ИФ* – имя фрейма, *ТФ* – тип фрейма, *ССЛ* – структура слотов, *Сатр* – структуры атрибутов, *Иатр_m* – имя атрибута, *МО_m* – множество определения, *ЗА_m* – значение атрибута.

Также, фрейм можно рассматривать как узел некоторой сети. Связи могут быть следующих типов: экземпляр-класс и класс-суперкласс.

Фреймовое представление знаний достаточно наглядно, подходит для представления иерархических знаний, хорошо сочетается с ООП, поэтому переход от представления знаний к программной реализации ИС представляется достаточно простым. К недостаткам фреймовой модели можно отнести сложность внесения изменений в иерархическую структуру данных.

Сочетание семантических сетей и фреймов позволяет минимизировать недостатки этих двух способов с сохранением достоинств.

4 способ – Применение фреймово-семантической модели представления знаний. А. Н. Швецов [51] предложил совместить фреймоподобные структуры с конструкциями концептуальных графов J. F. Sowa [52-54]. Основная конструкция фрейм-концепта (ФК) представлена формулой 2.7. Имя фрейма является уникальным идентификатором, позволяющим однозначно определить фрейм. Информация о применении ФК представляет собой описание возможных ситуаций использования ФК, сценариев поведения, особенностей выбора и т.п. в произвольной форме. Динамическое поведение компонентов или агентов предметной области описывает структура сценариев поведения (ССП), в которую включен блок выбора сценария (БВСЦ), позволяющий формировать альтернативные пути поведения данного фрейма.

ССЛ состоит из двух структур: структуры концептов (СК) и СА. СК содержит список фрейм-концептов, в некотором отношении вложенных или порожденных охватывающим ФК, тип этого отношения указывается в поле «тип концептуального отношения», т.е. отношение данного ИК_i к ФК, где ИК_i – имя i-го концепта. Для установления логической организации предметной области ФК соединяются в структуры концептуальных графов. Концептуальный граф (КГ) есть двудольный граф, имеющий два типа вершин: вершины концептов, или концептуальные вершины, и вершины концептуальных отношений (КО). Таким образом, Швецовым предлагается использовать фреймово-семантическое представление знаний, которая выглядит следующим образом:

$$ФК = \langle ИФ, ТФ, ИП, ССП, ССЛ \rangle, \quad (2.7)$$

$$ССЛ = \langle СК, Сатр \rangle, \quad (2.8)$$

$$СК = \{(ИК_1, КО_1), (ИК_2, КО_2), \dots, (ИК_n, КО_n)\}, \quad (2.9)$$

$$Сатр = \{(Иатр_1, МО_1, ЗА_1), (Иатр_2, МО_2, ЗА_2), \dots, (Иатр_m, МО_m, ЗА_m)\}. \quad (2.10)$$

где ИФ – имя фрейма, ТФ – тип фрейма, ИП – информация о применении, ССП – структура сценария поведения, ССЛ – структура слотов, СК – структуры концептов, Сатр – структуры атрибутов, ИК_n – имя концепта, КО_n – концептуальное отношение, Иатр_m – имя атрибута, МО_m – множество определения, ЗА_m – значение атрибута.

Достоинства фреймово-семантического подхода: эффективно реализует иерархическое представление данных, хорошо сочетается с ООП, техническая реализация данной модели представления знаний на уровне реляционной БД решена в работе [46; 47].

Сравнительный анализ рассмотренных моделей представления знаний представлен в таблице 2.2.

Таблица 2.2

Сравнение различных моделей представления знаний

Требования к модели	Продукции	Сем. сети	Фреймы	ФК и КГ Швецова
Наглядность	НЕТ	ДА	ДА	ДА

Представление иерархических данных	НЕТ	ДА	ДА	ДА
Простота добавления новых знаний	ДА	НЕТ	НЕТ	ДА
Согласованность с ООП	НЕТ	НЕТ	ДА	ДА
Использование UML	НЕТ	НЕТ	ДА	ДА
Описание ИА	НЕТ	НЕТ	НЕТ	ДА

Вывод по разделу

Выбор фреймово-семантической модели представления знаний дает следующие преимущества:

- согласованность фреймово-семантической модели с концепцией ООП обосновывает применение объектных языков программирования при разработке БЗ и минимизирует затраты на создание программного обеспечения. Задача технической реализации данной модели представления знаний на уровне БД решена в работе [46; 47];

- единая модель представления знаний, которая подходит, как будет показано в следующих разделах, для предметной области МППР и ИС.

На основе выбранной модели представления знаний в разделе 2.5.1 описывается концептуальная модель предметной области (КМПО) МППР. КМПО проектирования ИС описана в разделе 2.5.2.

2.4. Построение модели разработки информационной системы

2.4.1. Концептуальная модель предметной области мультиагентных процессов преобразования ресурсов

Основываясь на модели БП, выбранной в разделе 2.3, можно построить следующую фреймово-семантическую модель МППР (рисунок 2.2) [55-57].

При автоматизации процессов предприятия любому преобразователю соответствует функция обработки данных, однонаправленная или двунаправленная (генерация, прием, передача, изменение, удаление).



Рисунок 2.2. Семантическая сеть МППР

В работе [59] предложена классификация ресурсов с точки зрения использования. На рисунке 2.3 представлена эта классификация.



Рисунок 2.3. Классификация ресурсов по типам использования

Агент управляет объектами ППР, выполняя следующие действия: 1) анализирует внешние параметры (текущую ситуацию); 2) диагностирует ситуацию, обращается к базе знаний. В случае определения соответствующей ситуации агент пытается найти решение (сценарий действий) в базе знаний или выработать его самостоятельно; 3) вырабатывает (принимает) решение; 4)

определяет (переопределяет) цели; 5) контролирует достижение целей; 6) делегирует цели своим и чужим объектам процесса преобразования ресурсов, а также другим агентам; 7) обменивается сообщениями.

Архитектура агента МППР основана на гибридной архитектуре InteRRaP и представлена на рисунке 2.5.

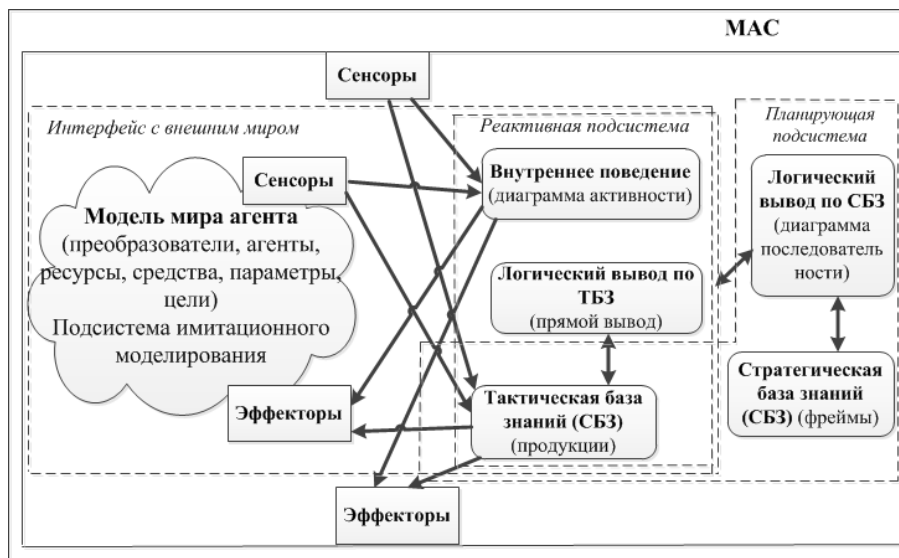


Рисунок 2.5. Архитектура агента МППР

Модель ОТС, представленная в виде модели МППР, показана на рисунке 2.6.

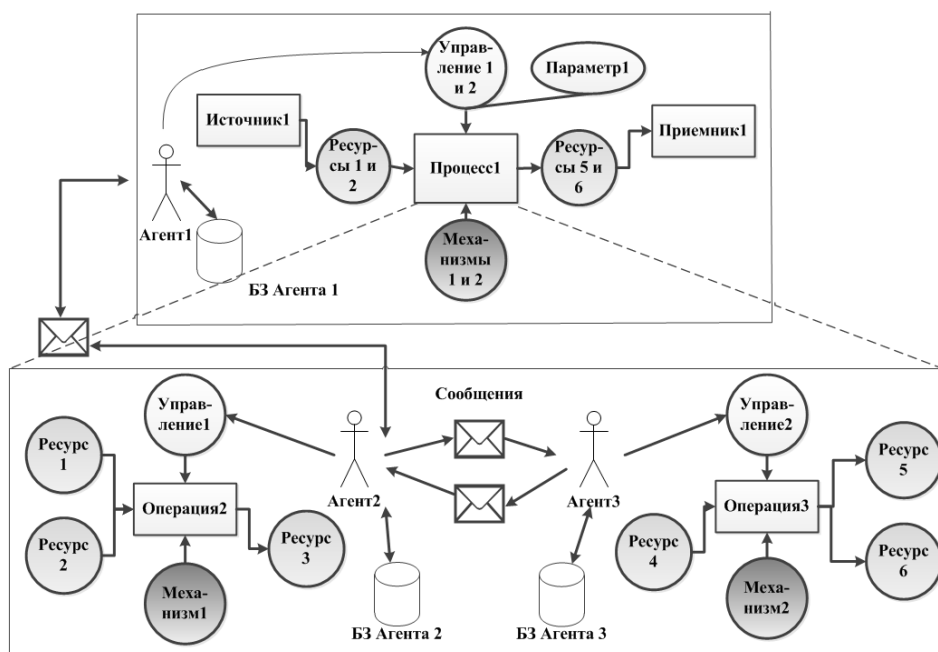


Рисунок 2.6. Модель ОТС

Определим свойства и методы фрейм-концептов (ФК) предметной области МППР. ФК перечисляются по мере их усложнения и согласно

принципу «от общего к частному». В скобках у ФК указывается название родительского ФК, если такой есть.

ФК «Ресурс»

Свойства:

- 1) идентификатор(ID_{RES}),
- 2) название ресурса (RES_Name),
- 3) тип ресурса ($kind$),
- 4) текущее значение ресурса (RES_t),
- 5) максимально возможное значение ресурса (RES_{max}),
- 6) начальное значение ресурса (RES_0),
- 7) конечное значение ресурса (RES_T),
- 8) цена единицы ресурса ($Cost$),
- 9) время начала моделирования (t_0),
- 10) единица измерения ресурса ($Metric_R$).

Методы:

- 1) суммарное приращение ресурса за интервал времени ($Hsum_{inc_t}$),
- 2) суммарное уменьшение ресурса за интервал времени ($Hsum_{dec_t}$),
- 3) стоимость ресурса ($Cost_sum$),
- 4) приращение ресурса в текущий момент времени (H_{inc}),
- 5) уменьшение ресурса в текущий момент времени (H_{dec}).

ФК «Команда» (Ресурс)

Свойства:

- 1) идентификатор(ID_Cmd),
- 2) название ресурса (Cmd_Name),
- 3) тип команды ($kind$),
- 4) отправитель ($Msender$),
- 5) получатель ($Mresiver$),
- 6) текст сообщения ($text$),
- 7) приоритет ($prior$),
- 8) признак обработки ($read$),

- 9) время создания (T_{create}),
- 10) время ожидания в очереди (T_{wait}).

ФК «Заявка» (Ресурс)

Свойства:

- 1) идентификатор (ID_Order),
- 2) название ресурса ($Order_Name$),
- 3) заказываемый объем работ ($count$),
- 4) выполненный объем работ ($real$),
- 5) признак блокировки заявки ($lock$),
- 6) имя элемента, обрабатывающего заявку ($Owner$),
- 7) имя блока, создавшего заявку ($Parent$),
- 8) приоритет заявки ($prior$),
- 9) время создания заявки (T_{create}),
- 10) время ожидания заявки в очереди (T_{wait}).

ФК «Средство» (Ресурс, Операция)

Свойства:

- 1) идентификатор (ID_Mech),
- 2) название ресурса ($Mech_Name$),
- 3) тип команды ($kind$),
- 4) текущее количество свободных средств ($Mech_t$),
- 5) всего средств ($Mech_{all}$),
- 6) время создания (T_{create}),
- 7) состояние средства ($Status$),
- 8) единовременные затраты ресурсов при начале преобразования (RES_{in}),
- 9) единовременные затраты ресурсов при окончании преобразования (RES_{out}),
- 10) расход ресурсов в единицу времени (RES_{use}),
- 11) единовременные затраты ресурсов при захвате средства другой операцией (RES_{lock}),
- 12) единовременные затраты ресурсов при освобождении (RES_{unlock}),
- 13) затраты ресурсов при возникновении и устранении поломки (RES_{other}),

- 14) периодичность возникновения поломки (T_{other}),
- 15) начальная цена единицы средства ($Cost$),
- 16) суммарное время использования средства (T_{mech_use}),
- 17) суммарное время простоя средства (T_{mech_stand}).

Методы:

- 1) действие по запуску средства в момент начала преобразования (Am_{in}),
- 2) действие по остановке средства в момент окончания преобразования (Am_{out}),
- 3) действие по выполнению преобразования (Am_{use}),
- 4) действие по остановке средства в момент прерывания преобразования (Am_{lock}),
- 5) действие по запуску средства в момент продолжения преобразования (Am_{unlock}),
- 6) действие по устранению поломки (Am_{other}),
- 7) производительность средства в единицу времени ($product$).

Операция является наиболее общим элементом МППР, остальные (источники, приемники, перекрестки) содержат усеченный набор свойств и методов, поэтому приведем только описания ФК «Операция». У ФК «Источник» есть только множество выходов. У ФК «Приемник» есть только множество входов. У ФК «Перекресток» есть модель его поведения.

ФК «Операция» (Преобразователь)

Свойства:

- 1) множество входов (IN),
- 2) множество выходов (OUT),
- 3) множество ресурсов, необходимых для прерывания операции (RES_{lock}),
- 4) множество ресурсов, необходимых для продолжения выполнения операции, остановленной в результате прерывания (RES_{unlock}),
- 5) цели операции ($Goal$),
- 6) средства преобразования ($MECH$),
- 7) состояние операции ($Status$),
- 8) длительность выполнения преобразования ($time$),

- 9) приоритет операции (*prior*),
- 10) тип приоритета (*kind_prior*),
- 11) признак запрета прерывания (*break_off*).

Методы:

- 1) функция, реализуемая операцией (*f*);
- 2) $C_a^{message}$ – условие наличия необходимых входных сообщений;
- 3) C_a^{order} – условие наличия необходимых входных заявок;
- 4) C_a^{in} – условие наличия необходимых входных ресурсов;
- 5) C_a^{out} – условие учета ограничений выхода;
- 6) C_a^{mech} – условие готовности необходимых средств;
- 7) C_a^{status} – условие готовности к исполнению;
- 8) C_a^{time} – условие запуска по времени.
- 9) $Action_{in}^{Message}$ – захват входного сообщения;
- 10) $Action_{in}^{Order}$ – захват входных заявок;
- 11) $Action_{in}^{RES}$ – захват входных ресурсов;
- 12) $Action_{in}^{MECH}$ – захват средств;
- 13) $Action_{out}^{Message}$ – формирование выходных сообщений;
- 14) $Action_{out}^{Order}$ – формирование выходных заявок;
- 15) $Action_{out}^{RES}$ – формирование выходных ресурсов;
- 16) $Action_{out}^{MECH}$ – освобождение захваченных средств.

ФК «Агент»

Свойства:

- 1) идентификатор (*ID_Agent*),
- 2) название агента (*Agent_Name*),
- 3) цели агента (*Gagent*),
- 4) приоритет агента (*prior*),
- 5) база знаний агента (*KVAg*),
- 6) количество входящих сообщений (*Messin_count*),

- 7) количество исходящих сообщений (*Messout_count*),
- 8) сценарий поведения (*SPA*),
- 9) множество управляемых объектов МППР (*Control_objects*),
- 10) множество агентов «начальников» (*AU*),
- 11) множество подчиненных агентов (*AD*).

Методы:

- 1) анализ мира (*Analize*),
- 2) диагностирование ситуаций (*Diagnost*),
- 3) поиск решения (*Search*),
- 4) обработка целей,
- 5) контроль достижения целей,
- 6) делегирование целей,
- 7) обмен сообщениями,
- 8) взаимодействие с БЗ и БД.

ФК «Параметр»

Свойства:

- 1) идентификатор (*ID_Param*),
- 2) название параметра (*Param_Name*),
- 3) текущее значение параметра (P_t),
- 4) начальное значение параметра (P_0),
- 5) описание параметра (*desc*), плановое значение (*plan*).

Методы:

- 1) функция вычисления параметра (f).

Таким образом, в данном подразделе описаны основные ФК КМПО предметной области МППР. В следующем разделе решается задача построения КМПО проектирование ИС.

2.4.2. Концептуальная модель предметной области информационных систем

Предлагаемая КМПО ИС позволяет показать структуру ИС и взаимосвязи между всеми ее составляющими (см. рисунок 2.7) [60; 61]. На первом уровне

фреймво-семантической сети находятся узлы, соответствующие программному обеспечению и базе данных ИС. Поскольку при моделировании и разработке ИС используют функциональный и объектно-ориентированный подходы (см. раздел 1.1.2), то она также содержит ФК элементов моделирования архитектуры ПО, включая ФК следующих диаграмм: функциональных (IDEF0), потоков данных (DFD) (рисунок 2.8) и диаграмм языка UML (прецедентов, последовательностей и классов, рисунки 2.9-2.11) [57; 62]. Эти стандарты не включают в себя описание конкретных реализаций операций, тем не менее, соответствующие ФК содержат методы, аналогичные методам ФК «Операция». Это позволяет сохранить данную информацию при переходе от модели МППР к модели ИС, а затем использовать ее при подготовке программных модулей.

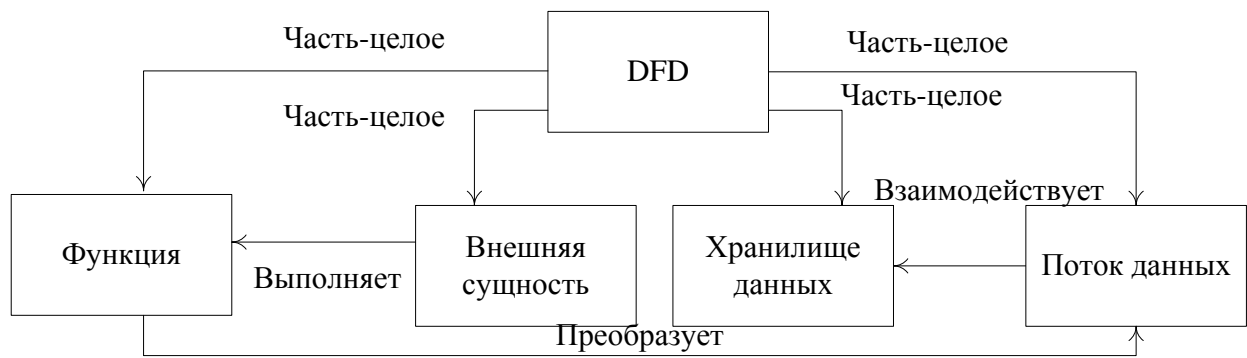


Рисунок 2.8. Семантика DFD-диаграммы

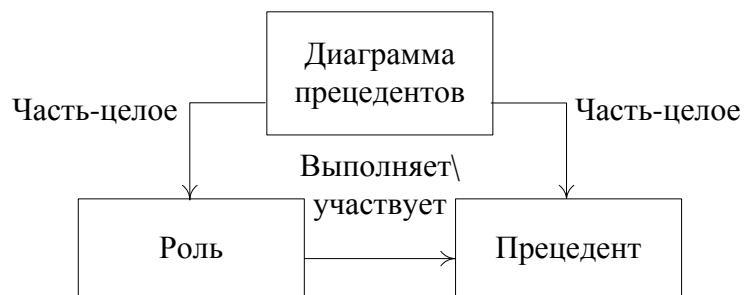


Рисунок 2.9 – Семантика диаграммы прецедентов

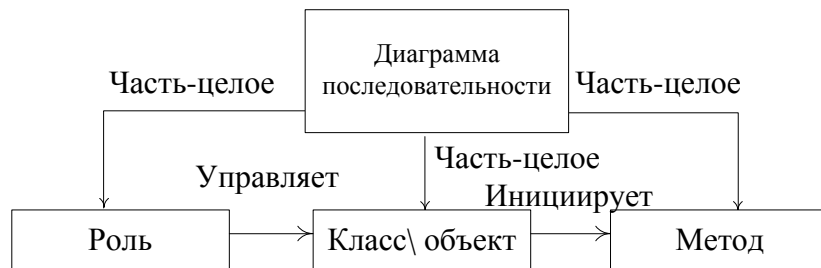


Рисунок 2.10. Семантика диаграммы последовательности

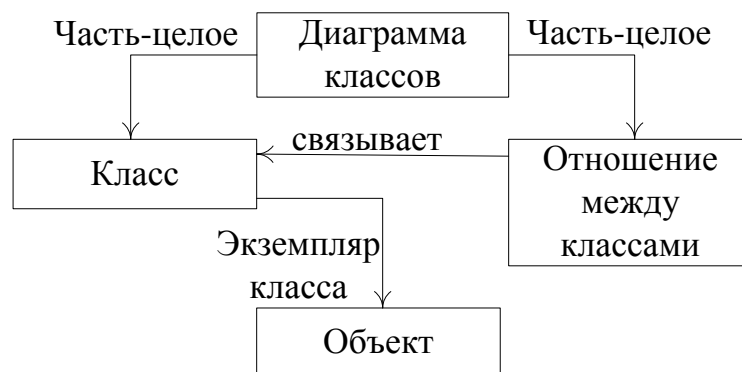


Рисунок 2.11. Семантика диаграммы классов

Определим структуру фрейм-концептов.

1. Фрейм-концепты DFD-диаграммы

1.1. ФК «Функция»

Свойства:

- 1) идентификатор(ID_F);
- 2) название (F_Name);
- 3) множество входов (IN);
- 4) множество выходов (OUT).

Методы:

- 1) функция преобразования (f).

1.2. ФК «Внешняя сущность»

Свойства:

- 1) идентификатор(ID_{Ex}),
- 2) название внешней сущности (Ex_Name),
- 3) цели внешней сущности (ExG),
- 4) приоритет внешней сущности ($Exprior$).

1.3. ФК «Хранилище данных»

Свойства:

- 1) идентификатор(ID_{DS}),
- 2) название хранилища данных (DS_Name),
- 3) структура хранилища данных ($DS_Structure$).

1.4. ФК «Поток данных»

Свойства:

- 1) идентификатор(ID_{DF}),
- 2) название потока данных (DF_Name),
- 3) тип потока данных= $\{Select, Insert, Delete, Update, Unknown\}$ (DF_kind),
- 4) текущее значение потока данных (DFt),
- 5) максимально возможное значение потока данных ($Dfmax$),
- 6) начальное значение потока данных ($DF0$),
- 7) конечное значение потока данных (DFT),

- 8) цена единицы потока данных (DF_Cost),
- 9) единица измерения потока данных (DF_Metric_R).

2. Фрейм-концепты диаграммы прецедентов

2.1. ФК «Роль»

Свойства:

- 1) идентификатор(ID_R),
- 2) название роли (R_Name),
- 3) цели роли (RG),
- 4) приоритет роли ($Rprior$).

2.2. ФК «Прецедент»

Свойства:

- 1) идентификатор(ID_UC);
- 2) название (UC_Name).

Методы:

- 1) функция преобразования (UC_f),
- 2) условие запуска (UC_Ca).

3. Фрейм-концепты диаграммы классов

3.1. ФК «Класс»

Свойства:

- 1) идентификатор(ID_class),
- 2) название ($Class_Name$),
- 3) другие свойства, определяются предметной областью.

Методы:

Методы класса определяются предметной областью.

3.2. ФК «Объект»

Представляет собой экземпляр соответствующего класса, его свойства и методы определяются свойствами и методами соответствующего класса.

3.3. ФК «Отношение между классами»

Возможны следующие типы отношений [13]:

- ассоциации – произвольная взаимосвязь;

- обобщение – взаимосвязь между классом-родителем и классом-потомком;
- агрегация – взаимосвязь между классом-контейнером и классом-частью;
- композиция – агрегация, при которой с уничтожением класса-контейнера уничтожаются все его классы-части;
- зависимость – взаимосвязь между классами, при которой одному или нескольким классам требуются другие классы для их спецификации или реализации;
- реализация – взаимосвязь между классами, при которой один класс представляет некоторую спецификацию, а другой – его реализацию.

Свойства:

- 1) идентификатор (ID_{rl}),
- 2) название (Rl_Name),
- 3) тип связи (Rl_type),
- 4) множество начальных классов ($Beg_classes$),
- 5) множество конечных классов ($End_classes$).

Предлагаемый метод ППР в области моделирования и разработки ИС должен решать задачу перехода от КМПО МППР к КМПО ИС. Этот метод изложен в следующем разделе.

2.5. Метод разработки информационных систем

Рассмотрим, как КМПО МППР преобразуется в КМПО ИС (ФК DFD-диаграммы и ФК UML-диаграмм). Для описания семантики переходов между различными моделями БП и ИС используется диаграмма состояния объекта (стандарта IDEF5) [55].

Ресурс (RES).

Ресурс в ИС может представлять собой переменную, т.е.

$$RESin_i \rightarrow VAR_i, RESout_k \rightarrow VAR_k, \quad (2.11)$$

где $RESin_i$ – i -ый входной ресурс,

VAR_i – i -ая переменная,

$RESout_k$ – k -ый выходной ресурс.

Далее, двигаясь по семантической сети, выбираем вершины с отношением Вид-род и предком Простой тип или Пользовательский тип. Таким образом, можно определить нужный тип для хранения ресурса. В более сложных случаях ресурс может представлять собой сложную структуру: таблицу (на уровне БД) и/или массив (на уровне переменных кода). Эта семантика перехода ресурса приведена на рисунке 2.12. Семантика перехода ресурса в объекты диаграмм показана на рисунке 2.13.

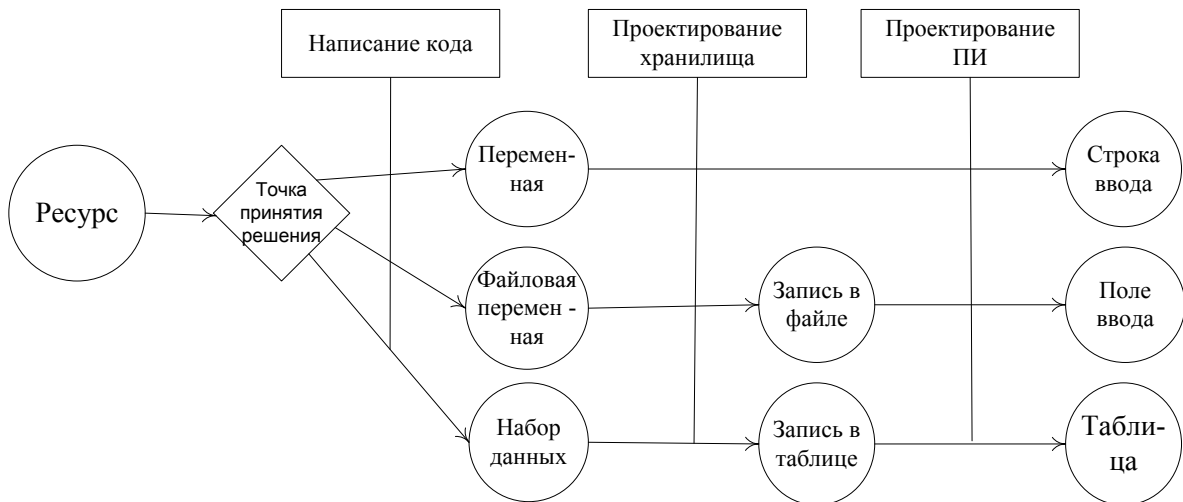


Рисунок 2.12. Семантика перехода ресурса в элементы ИС

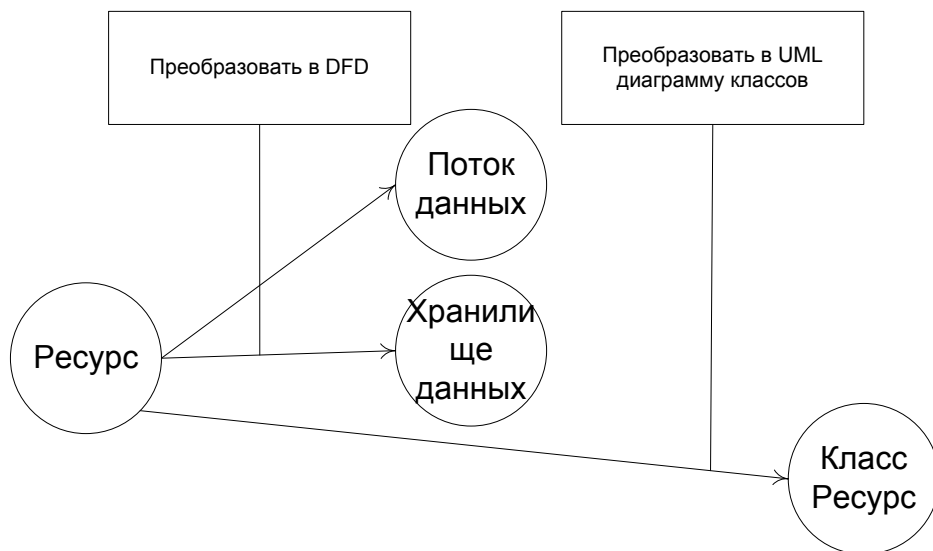


Рисунок 2.13. Семантика перехода ресурса в объекты диаграммы

Сценарий перехода ресурса можно представить следующей схемой (рисунок 2.14):

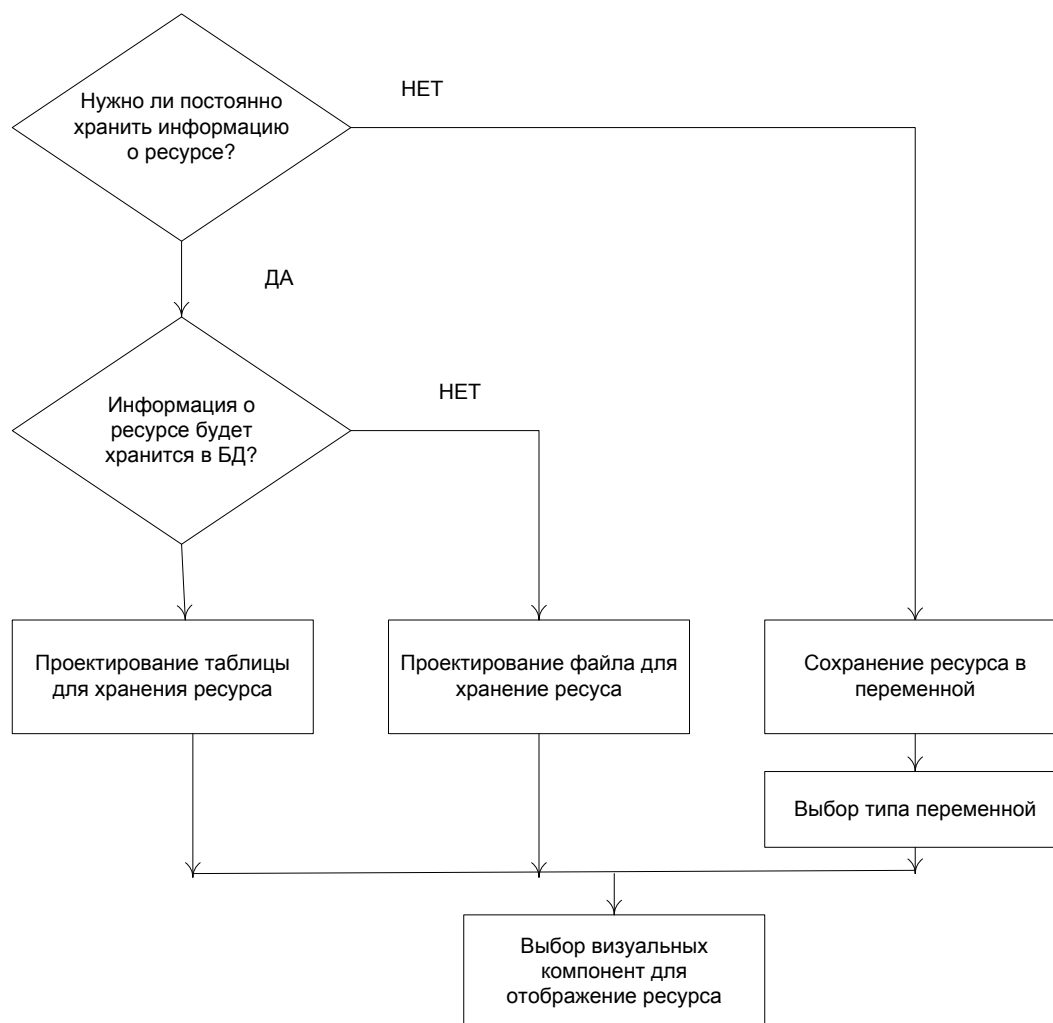


Рисунок 2.14. Блок-схема перехода ресурса

Преобразователь

К преобразователю относятся процессы, источники, приемники и перекрестки. Преобразователь как функция преобразования входных ресурсов в выходные может быть реализована в виде функции (ПО), которая выполняет обработку данных в памяти или на уровне файлов, или хранимой процедуры (ХП) базы данных (БД), т. е.

$$Tr_i \rightarrow Func_i, Tr_i \rightarrow StoredProc_i. \quad (2.12)$$

где Tr_i – преобразователь;

$Func_i$, – функция ПО;

$StoredProc_i$, – хранимая процедура.

При необходимости, следует предусмотреть наличие элементов ПИ для определения условий начала преобразования. Входными параметрами для функции или ХП будут входные ресурсы, а выходными параметрами –

выходные ресурсы. Таким образом, определив на 1-м этапе ресурсы, тем самым задаются типы параметров функций или ХП. В процессе интеллектуального моделирования и разработки ПО можно предоставить возможность описания алгоритма выполнения операции на алгоритмическом языке или T-SQL.

Семантика перехода преобразователя приведена на рисунке 2.15.

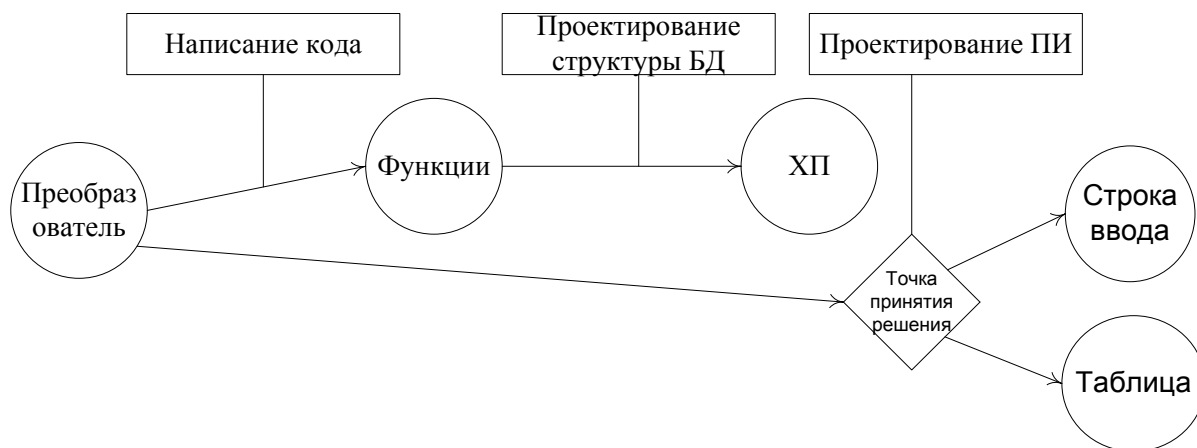


Рисунок 2.15. Семантика перехода преобразователя

Семантика перехода ресурса в объекты диаграмм показана на рисунке 2.16.

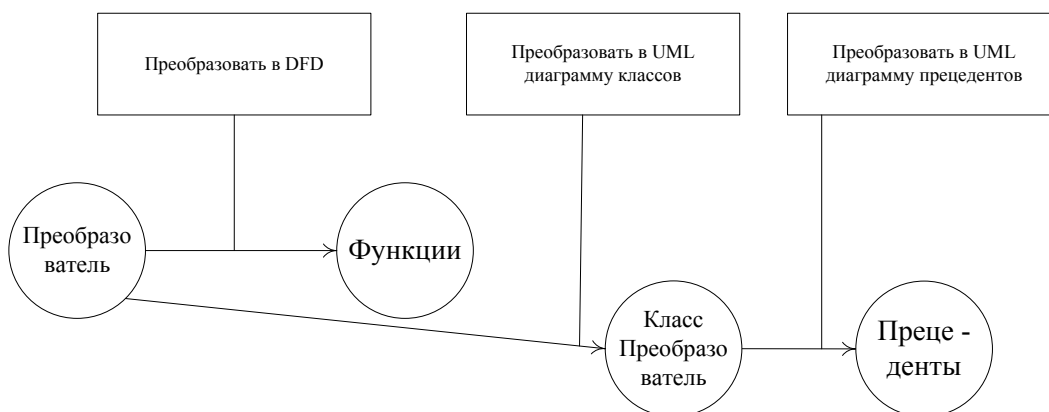


Рисунок 2.16. Семантика перехода преобразователя в объекты диаграммы

Переход от КМПО МППР к КМПО ИС при работе с преобразователем определяется результатом перехода соответствующих ресурсов.

Сценарий перехода преобразователя к элементам ИС показан на рисунке 2.17.

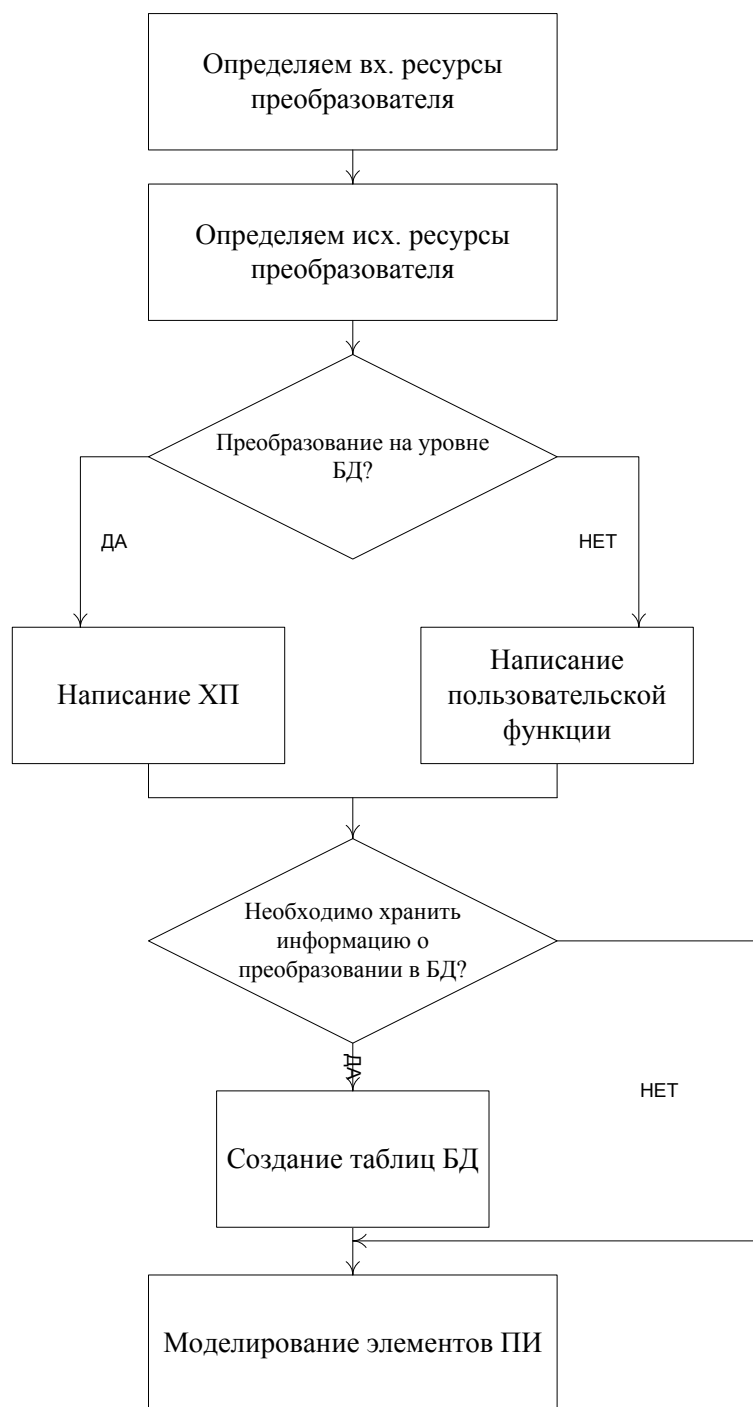


Рисунок 2.17. Блок-схема перехода преобразователя

Средства (МЕСН)

Средствами выполнения операций могут быть различные технические устройства, которыми оборудовано рабочее место человека, например, контроллеры, датчики, компьютер, принтер, сканер. Информация об их характеристиках, полученная в процессе проектирования идет в раздел технические требования к ПО. В некоторых ситуациях оператора этих устройств тоже можно рассматривать как средство. При моделировании и

разработке ИС необходимо учитывать характеристики процессов обработки и передачи данных, т.к. в конечном итоге они оказывают влияние на характеристики основного процесса.

При разработке человеко-машинных программных комплексов при решении вопросов защиты информации средства устанавливаются в соответствие определенные роли и назначаются права доступа к тем или иным объектам и функциям ИС. Таким образом, наблюдается аналогия с ресурсом. Поэтому за основу схемы проектирования средств взята схема проектирования ресурса.

Параметры (P)

Некоторая демонстрация пользователю характеристик протекания процесса или операции. В простейшем случае – это компонент «полоса прогресса», демонстрирующий как долго процесс еще будет работать. Также это может быть некоторое значение, вычисляемое по формуле, например какая-то характеристика ресурса. Следовательно, пользователь должен определить формулу и способ отображения параметра (график, текстовое поле), т. е.

$$P_k \rightarrow \langle Component_k, F_k \rangle, \quad (2.13)$$

где P_k – k -ый параметр,

$Component_k$ – k -ый визуальный компонент ПИ,

F_k – k -ая функция ПО.

Семантика перехода параметра представлена на рисунке 2.18.

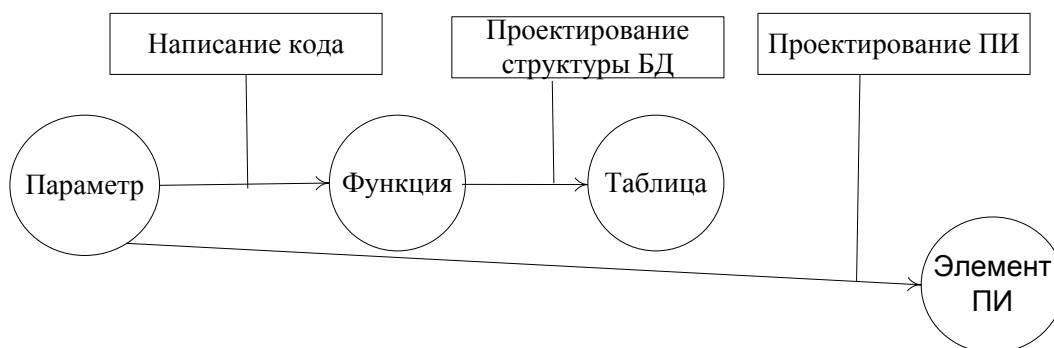


Рисунок 2.18. Семантика перехода параметра

Рассмотрим сценарий перехода параметра (рисунок 2.19).

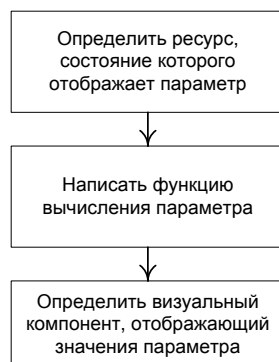


Рисунок 2.19. Блок-схема перехода параметра

Агенты (*Agent*)

Агенту соответствует модель ЛПР, имеющая сложную структуру. При ее построении используют подходы искусственного интеллекта (например, в основе модели может лежать алгоритм или сценарий поведения, реализованный в виде ЭС; она включает: машину логического вывода, базу знаний, правила и рабочую память). С точки зрения ИС, агент представляет собой программную сущность, у которой, при необходимости, имеется ПИ, функции, описывающие сценарий работы агента, и таблица для хранения БЗ агента. Возможна реализация сценария на уровне БД в виде ХП. Семантика перехода агента в элементы ИС приведена на рисунке 2.20.

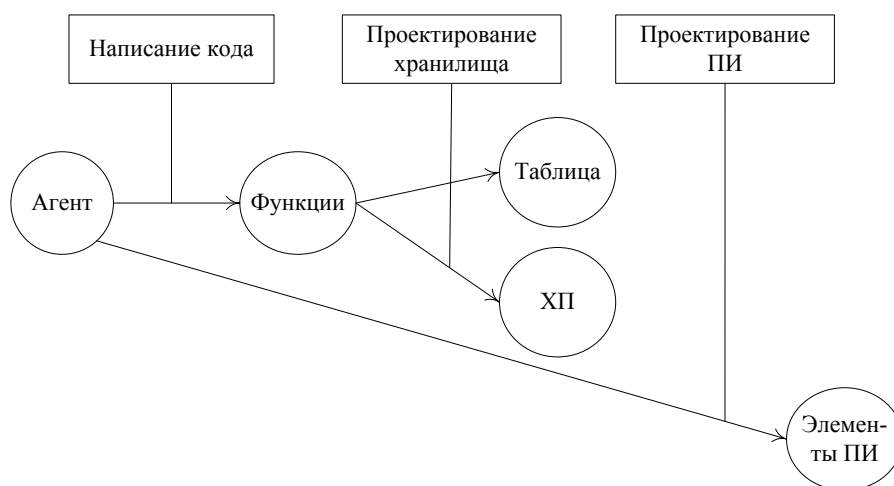


Рисунок 2.20. Семантика перехода агента в элементы ИС

Семантика перехода агента в объекты диаграмм приведена на рисунке 2.21. С помощью диаграммы последовательности моделируется сценарий поведения агента.

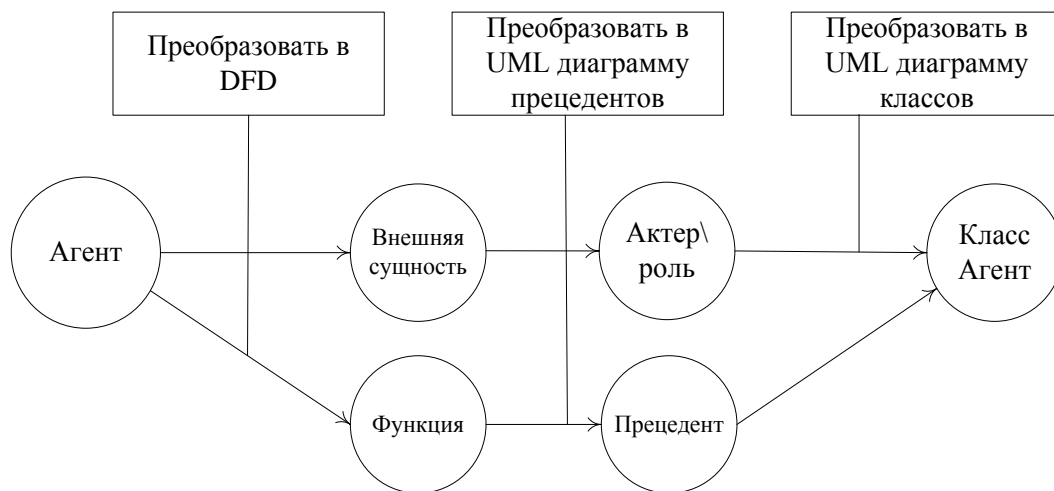


Рисунок 2.21. Семантика перехода агента в объекты диаграмм

Рассмотрим сценарий перехода агента в объекты ИС (рисунок 2.22).



Рисунок 2.22. Блок-схема перехода агента

Далее приводится метод ППР разработки ИС предметной области МППР, состоящий из следующих этапов [57]:

1. Разработка ИС начинается с обследования предметной области и построения имитационной модели МППР «как есть».
2. На втором этапе проводятся имитационные эксперименты с моделью «как есть» с целью выявления «узких мест» в организации процессов. По результатам моделирования строится модель МППР «как будет».

3. На третьем этапе осуществляется построение модели ИС на основании данных из модели МППР.

3.1. Каждая операция из модели МППР, которую необходимо автоматизировать в ИС, преобразуется в функцию DFD-диаграммы $\forall Op_i : Op_i \rightarrow F_{DFD_i}$. На диаграмме классов создается базовый класс операций, для каждой операции – экземпляр базового класса.

3.2. Все ресурсы, используемые в автоматизируемых операциях, преобразуются в потоки данных DFD-диаграммы, причем входные ресурсы i -ой операции становятся входными потоками i -ой функции DFD-диаграммы $\forall Re s_i : Re s_i \in InOp_i \quad Re s_i \rightarrow DataFlowIn(Op_i)$, а выходные ресурсы – выходными потоками i -ой функции DFD-диаграммы $\forall Re s_i : Re s_i \in OutOp_i \quad Re s_i \rightarrow DataFlowOut(Op_i)$. На диаграмме классов создается базовый класс ресурса. Для каждого ресурса – экземпляр базового класса.

3.3. Для всех ресурсов модели необходимо создать хранилища данных $Create(DataStore_i)$ на DFD-диаграмме. На UML-диаграмме классов создается базовый класс для хранилища данных.

3.4. Все агенты из модели МППР, которые будут реализованы программно, преобразуются во внешние сущности DFD-диаграммы $\forall Agent_i : Agent_i \rightarrow ExtrEss_i$. На диаграмме классов создается базовый класс агента. Для каждого агента – экземпляр базового класса. $\forall Agent_i : Agent_i \rightarrow Class(Agent_i)$. На основании данных из DFD-диаграммы создаются диаграммы прецедентов. Каждая внешняя сущность преобразуется в актера соответствующей диаграммы прецедентов $\forall ExtrEss_i : ExtrEss_i \rightarrow Actor_i$, а связанные с ней функции – в прецеденты $\forall F_i : F_i \rightarrow Case(Actor_i)$.

3.4.1. Преобразование агента

Рассмотрим на примере агента с одним правилом («если» $a > b$, «то» $a = a - b$). Элементы памяти, необходимые для хранения переменных, преобразуются в хранилища данных DFD-диаграммы. Правила «если» и «то» – в операции. В результате, получаем DFD-диаграмму, представленную на рисунке 2.23.

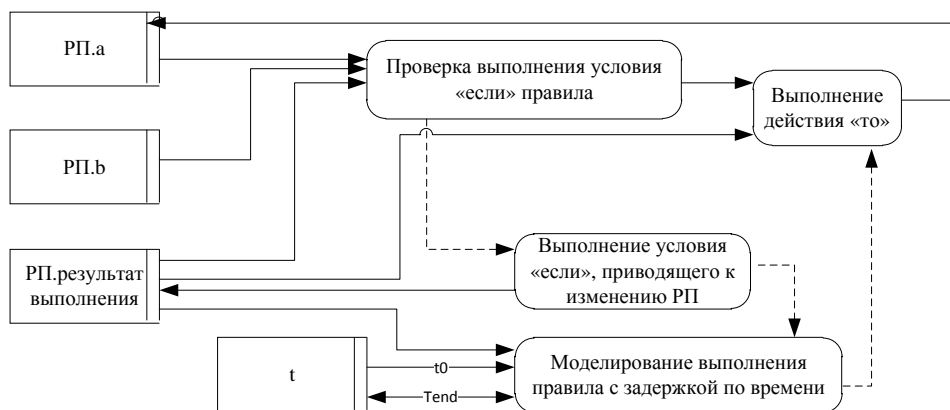


Рисунок 2.23. Пример DFD-диаграммы для реактивного агента с одним правилом

3.4.2. Описание правил агента используется для построения диаграммы прецедентов, т.е. каждое правило переходит в прецедент.

3.4.3. Формулы, содержащиеся в условиях «если» и «то» правил агента переходят в описание метода соответствующего класса. На рисунке 2.24 представлена диаграмма поиска решения ИА проектирования ИС при конвертации агента на продукциях. Следует отметить, что:

- ресурсы, средства и заявки представляют собой рабочую память;
- на шаге 2 если ни одна ситуация не диагностирована, то переход на шаг 11.

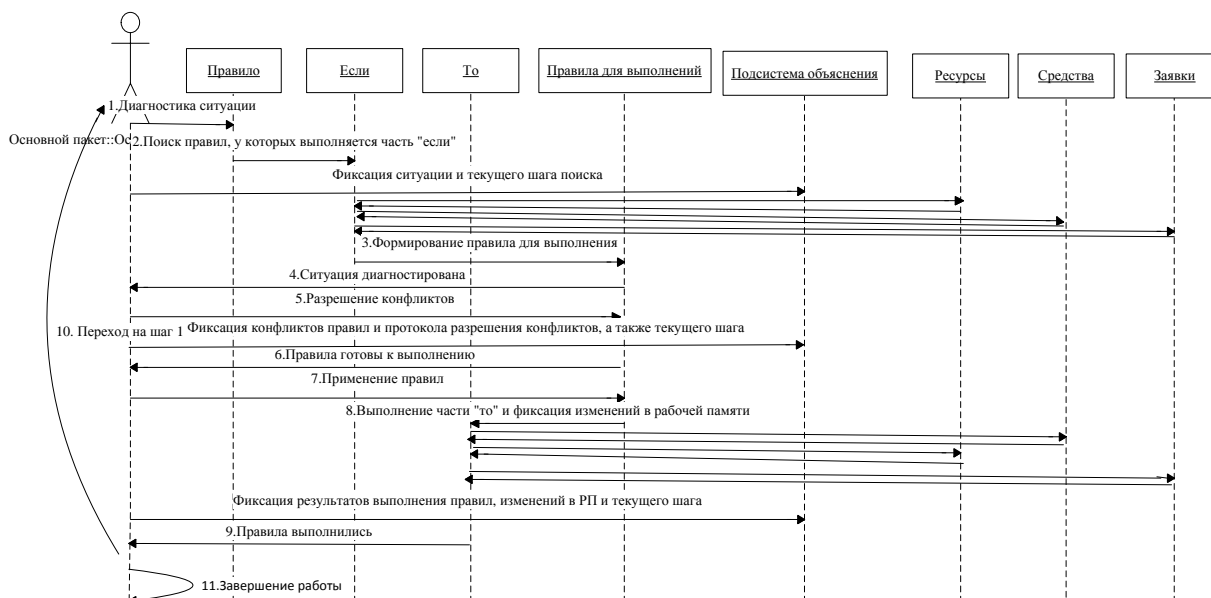


Рисунок 2.24. Диаграмма поиска решения ИА проектирования ИС при конвертации агента на продукциях

База знаний ИА проектирования ИС представляет собой описание объектов МППР и ИС.

3.5. На основании данных из DFD-диаграммы создаются диаграммы прецедентов. Каждая внешняя сущность преобразуется в актера соответствующей диаграммы прецедентов $\forall Ex_{DFD}^i \rightarrow Actor_{UML}^i$, а связанные с ней функции – в прецеденты $\forall F_{DFD}^i : F_{DFD}^i(Ex_{DFD}^i) \rightarrow Case_{UML}^i(Actor_{UML}^i)$.

3.6. Атрибуты классов, соответствующих внешним сущностям, позволяют определить структуру таблиц ER-диаграммы.

4. Доработка системы разработчиками, построение диаграмм последовательности и моделирование пользовательского интерфейса.

5. Решение вопроса о размещении экземпляров концептов предметной области по базам знаний (KBАg) агентов. Для решения этого вопроса рассмотрим задачу размещения с дискретным пространством решений [64]. Ее постановка представлена ниже.

найти

$$\min z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.14)$$

при ограничениях

$$\sum_{j=1}^n a_{ij} x_{ij} \geq 1, i=1, \dots, m \quad (2.15)$$

$$x_{ij} \in (0,1), i=1, \dots, m; j=1, \dots, n \quad (2.16)$$

где n – количество агентов,

m – количество экземпляров концептов предметной области,

c_{ij} – коэффициент, показывающий величину затрат на размещение i -го экземпляра концепта у j -го агента,

$$x_{ij} = \begin{cases} 1, & \text{если } i - \text{ый экземпляр концепта} \\ & \text{расположен у } j - \text{го агента} \\ 0, & \text{иначе} \end{cases},$$

a_{ij} – коэффициент, определяющий потребность j -го агента в i -м концепте,

$$a_{ij} = \begin{cases} 1, \text{ если } i - \text{ый экземпляр концепта} \\ \quad \text{нужен } j - \text{му агенту} \\ 0, \text{ иначе} \end{cases}.$$

Таким образом, необходимо размесить экземпляры концептов по базам знаний (*KBAg*) агентов с минимальными затратами. Коэффициенты c_{ij} определяются экспертами и позволяют линейно упорядочить агентов по стоимости размещения экземпляров концептов с учетом стоимости разработки распределенной ИС и ее эксплуатации. Для этого могут применяться методы экспертных оценок [65]. В работе данная задача не рассматривается.

Постановка, приведенная для задачи размещения концептов по базам знаний (*KBAg*) агентов, имеет свою особенность. В имитационной модели БП имеется информация о том, какому агенту какие экземпляры концептов необходимы, и данная информация используется в ограничении задачи. По сути, известны допустимые варианты размещения концептов, среди которых нужно выбрать минимальный по затратам. Если потребности агентов в концептах достаточно обособлены друг от друга, то число возможных вариантов решения существенно сокращается по сравнению с полным перебором возможных вариантов.

Поскольку рассматриваемая задача решается в рамках автоматизации бизнес-процессов, то в сравнении с классической задачей необходимо учитывать, что действия агентов являются составной частью автоматизируемого бизнес-процесса и должны укладываться в определенные временные рамки $T_{\text{БП}}^{\text{max}}$ – максимальное время выполнения бизнес-процесса. Следовательно, к ограничениям необходимо добавить ограничение на время выполнения бизнес-процесса ($T_{\text{БП}}$), которое в общем случае будет иметь вид:

$$T_{\text{БП}} \leq T_{\text{БП}}^{\text{max}} \quad (2.17)$$

Рассмотрим решение этой задачи для случая разработки системы ведения реестров. Пусть необходимо разместить 5 (пять) реестров между 3 (тремя) агентами. Законодательно, существует ограничение на выполнение операции –

3 дня. В случае нарушения сроков могут быть предъявлены существенные штрафы. Особенность данной системы состоит в том, что не все агенты равноправны. Один из них, центральный офис, должен хранить у себя все 5 реестров, а остальным двум агентам нужен доступ только к тем реестрам, с которыми они работают. Экспертные оценки затрат на размещение концептов предметной области по базам знаний приведены в таблице 2.3.

Таблица 2.3

Экспертные оценки затрат на размещение

концепты/агенты	1	2	3
1	2	1	1
2	2	1	1
3	2	1	1
4	2	1	1
5	2	1	1

Постановка задачи:

$$\min z = \sum_{i=1}^5 \sum_{j=1}^n c_{ij} x_{ij}$$

при ограничении:

$$a_{11} + a_{21} + a_{31} + a_{41} + a_{51} \geq 1$$

$$a_{12} + a_{22} \geq 1$$

$$a_{23} + a_{33} + a_{43} \geq 1$$

$$x_j \in (0,1), j=1, \dots, 5$$

$$T_{\text{БП}} \leq 3$$

Для наглядности, представим ограничения в табличной форме (таблица 2.4): на пересечении *i*-ой строки и *j*-го столбца стоит 1, если $a_{ij} = 1$.

Таблица 2.4

Представление значений a_{ij} в виде таблицы

агенты/концепты	1	2	3	4	5
1	1	1	1	1	1
2	1	1	0	0	0
3	0	1	1	1	0

Рассмотрим возможные варианты решений.

1 вариант – размещение всех экземпляров концептов в центральном офисе.

В таблице 2.5 представлены значения x_{ij} и целевой функции z для 1-го варианта.

Таблица 2.5

Решение для 1-го варианта

агенты/концепты	1	2	3	4	5	z
1	1	1	1	1	1	10
2	0	0	0	0	0	
3	0	0	0	0	0	

Все остальные варианты размещения приводят к дублированию концептов предметной области, поскольку по условию у центрального офиса обязательно должны быть все реестры. Для примера рассмотрим несколько вариантов размещения с дублированием.

2 вариант. В таблице 2.6 представлены значения x_{ij} и целевой функции z для 2-го варианта.

Таблица 2.6

Решение для 2-го варианта

агенты/концепты	1	2	3	4	5	z
1	1	1	1	1	1	14
2	1	1	0	0	0	
3	0	0	1	1	0	

3 вариант. В таблице 2.7 представлены значения x_{ij} и целевой функции z для 3-го варианта.

Таблица 2.7

Решение для 3-го варианта

агенты/концепты	1	2	3	4	5	z
1	1	1	1	1	1	14
2	1	0	0	0	0	
3	0	1	1	1	0	

4 вариант. В таблице 2.8 представлены значения x_{ij} и целевой функции z для 4-го варианта.

Таблица 2.8

Решение для 4-го варианта

агенты/концепты	1	2	3	4	5	z
-----------------	---	---	---	---	---	---

1	1	1	1	1	1	12
2	1	1	0	0	0	
3	0	0	0	0	0	

Современный уровень развития информационных технологий позволяет реализовать такую информационную систему с соблюдением временных ограничений на БП.

Поскольку специфика БП такова, что длительные простои в работе ИС могут привести к существенным штрафам, необходимо оценить уровень надежности системы.

Непосредственный расчет вероятностей отказов и сбоев в сложной и многокомпонентной распределенной системе не представляется возможным, хотя бы в связи с отсутствием необходимых данных о надежности компонент. Подобные расчеты проводятся разве что в авиастроении, но и там определяющим фактором в оценке надежности является опыт эксплуатации. Заметим, что все эти оценки носят вероятностный характер, задается лишь некоторый допустимый предел вероятности отказа. Они никак не могут гарантировать отсутствия сбоев.

Эксплуатационная надежность серверов и аппаратуры ИС оценивается обычно суммарными временами простоя в течение календарного года. Эти данные фиксируются в логах и могут быть легко проанализированы. В настоящее время аппаратная часть ИС, а также операционная система и специализированное ПО достаточно надежны. Для продукции различных фирм фиксируются перебои в функционировании с продолжительностью от десятков минут и до нескольких часов в течение года из-за внутренних аппаратных и системных проблем [66]. Однако опыт работы реальных распределенных ИС показывает, что перебои, связанные с обменом данными между агентами, могут составлять до нескольких рабочих дней.

Нештатные и незапланированные простои в работе ИС, естественно, приводят к убыткам. Поэтому интерес представляет не просто вероятностные характеристики, а привязка их к возможным финансовым и прочим потерям. В нашей задаче, оказалось, достаточно затруднительно построить целевую

функцию, которая адекватно учитывала бы юридический и финансовый риск, связанный с простоем в работе, превышающем нормативно допустимый.

Для анализа связи между величиной надежности и продолжительностью простоев в работе [67] приводится следующая таблица (см. таблицу 2.9).

Таблица 2.9

Классификация систем по уровню надежности

Коэффициент готовности, Kr	Максимальное время простоя в год	Тип системы
0,99	3,5 сут.	Обычный
0,999	8,5 ч.	Высокой надежности
0,9999	1 ч.	Отказоустойчивый
0,99999	5 мин.	Безотказный

Фактически эти результаты можно получить в простой дискретной модели – схеме Бернулли. Отказы (в схеме Бернулли называются «успехами») характеризуются величиной Kr и принимаются за параметр p . «Испытанием» будет временная единица (сутки или час.), а длиной последовательности испытаний – количество временных единиц в календарном году. При этом продолжительность простоя оценивается как математическое ожидание ($N \cdot p$) числа «успехов» в схеме Бернулли.

Действительно, $N = 365 \cdot 24 = 8760$ часов, тогда $N \cdot p = \frac{8760}{100} = 87,5$ ч. Или 3,65 сут., что и стоит в строке таблицы для обычной системы. Для системы высокой надежности $N \cdot p = \frac{8760}{1000} = 8,75$ ч.

Как известно, распределение Бернулли таково, что «успехи» в серии могут быть не только изолированы «неудачами», но и образовывать кластеры, т. е. идущие подряд цепочки «успехов». В очень многих прикладных задачах, в частности и в рассматриваемой, интерес представляют вероятности того, что кластеры будут иметь определенную длину. Если периоды суточного простоя разделены промежутками нормального функционирования, то такой уровень надежности представляется приемлемым. Однако кластер длиной более 3 суток – неприемлем.

Рассмотрим схему Бернулли, в которой каждый рабочий день рассматривается как испытание, а событием является возможный перерыв в работе в течение этого рабочего дня. Параметр p – вероятность перерыва в работе в этот день. Обычная схема Бернулли дает возможность вычислить вероятность k «успехов» в серии N испытаний.

$$P_N(k) = C_N^k * p^k * (1 - p)^{N-k} \quad (2.18)$$

Однако более интересной является задача о вероятности нескольких идущих подряд «успехов». Таким образом моделируется длительный простой системы на протяжении нескольких рабочих дней подряд, что позволяет точнее оценить риск длительного прерывания в обслуживании клиентов.

В литературе по теории вероятностей данная задача многократно рассматривалась в различных вариантах постановки, например, [68]. Следует отметить, что она допускает точное аналитическое решение [69]. В то же время эта задача имеет достаточно широкую применимость, например, при изучении надежности компьютерных сетей [70].

В расчетах была использована формула для подсчета вероятности того, что длина кластера l_N в последовательности длиной N превзойдет некоторое критическое значение m [71]:

$$P(l_N \geq m) = \sum_{j=1}^{\lfloor \frac{N}{m} \rfloor} (-1)^{j+1} (p + (\frac{N-jm+1}{j})(1-p)) C_{N-jm}^{j-1} p^{jm} (1-p)^{j-1} \quad (2.19)$$

Использование данной формулы при малых значениях вероятностей может приводить к существенным погрешностям. Поэтому вычисления проводились в целочисленном режиме пакета Maple. Благодаря этому, погрешность округления проявилась только при переводе результирующих значений из рациональной дроби в десятичную.

Результаты приведены в таблице 2.10.

Таблица 2.10

**Решение задачи о вероятности последовательности серии «успехов»,
идущих подряд (2 дня и 3 дня)**

p	$\frac{1}{30}$	$\frac{1}{50}$	$\frac{1}{100}$	$\frac{1}{500}$	$\frac{1}{1000}$
$P(l_{365} \geq 2)$	0,32	0,133	0,035	0,015	0
$P(l_{365} \geq 3)$	0,013	0,0022	0,0004	0,000	0

Таким образом, в рамках рассмотренной модели значение для вероятности отказа порядка 0.01 ($r = 0.99$) следует считать как имеющим высокий риск, уменьшить который для приемлемого значения можно лишь для системы высокого уровня надежности, т. е. $Kr = 0.999$. Следовательно, снижение суммарного простоя с нескольких суток до нескольких часов соответствует повышению надежности на порядок.

Простейшим решением данной проблемы является резервирование (т. н. резервирование замещением с ненагруженным резервом). Пусть в случае отказа основной системы подключается резервная, значительно более дешевая в эксплуатации и, соответственно, менее надежная.

Расчет надежности в данном случае можно проводить в схеме гипотез, где гипотезами будут соответственно безотказная работа основной системы и ее отказ:

$$\sum_{k=1}^l P(A|H_k) * P(H_k) = P(A), \quad (2.20)$$

где

l – количество гипотез,

$P(A/H_k)$ – условная вероятность события A при справедливости H_k

$P(H_k)$ – априорная вероятность гипотез

В нашем примере возьмем за вероятности отказов соответственно 0.1 (резервная система) и 0.01 (основная система). Тогда по формуле полной вероятности получим:

$$0.99 + 0.01 * 0.9 = 0.999$$

Очевидно, что система с ненагруженным резервом и надежным переключением дает повышение общей надежности на порядок против отсутствия резервирования.

На рисунке 2.25 представлена схема БП в том случае, когда БЗ расположена в центре. В этом случае:

$$T_{\text{БП}} = t_{\text{обработки}}_{\text{A1}} + t_{\text{обработки}}_{\text{Центр}} + t_{\text{передачи}}_{\text{Центр-A1}}, \quad (2.21)$$

где

$t_{\text{обработки}}_{\text{A1}}$ – время работы агента1, включая человеко-машинное участие;

$t_{\text{обработки}}_{\text{Центр}}$ – время работы Центра, включая человеко-машинное участие;

$t_{\text{передачи}}_{\text{Центр-A1}}$ – время передачи результатов работы из Центра агенту1.

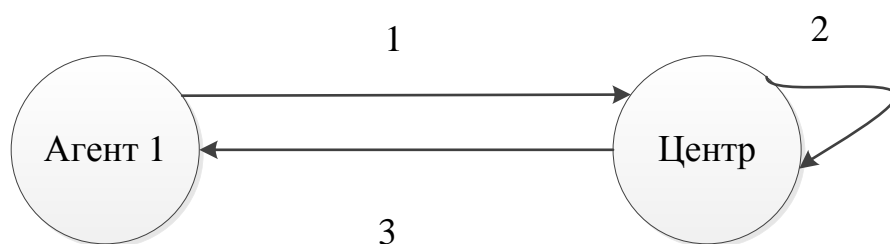


Рисунок 2.25. Схема БП в случае расположения БЗ в Центре

Рассмотрим теперь возможные варианты резервирования системы.

1 вариант. Размещение концептов, необходимых для работы агента у него. Этот вариант представлен на рисунке 2.26. В этом случае:

$$T_{\text{БП}} = t_{\text{обработки}}_{\text{A1}} + t_{\text{передачи}}_{\text{A1- Центр}} + t_{\text{обработки}}_{\text{Центр}} + t_{\text{передачи}}_{\text{Центр-A1}}, \quad (2.22)$$

где

$t_{\text{обработки}}_{\text{A1}}$ – время работы агента1, включая человеко-машинное участие;

$t_{\text{передачи}}_{\text{A1- Центр}}$ – время передачи результатов работы из агента1 в Центр;

$t_{\text{обработки}}_{\text{Центр}}$ – время работы Центра, включая человеко-машинное участие;

$t_{\text{передачи}}_{\text{Центр-A1}}$ – время передачи результатов работы из Центра агенту1.

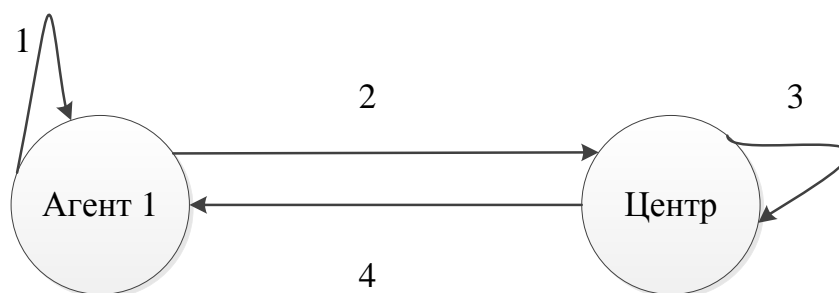


Рисунок 2.26. Схема БП в случае размещения БЗ в Центре, дублирование концептов у агентов

В случае отсутствия 2-3 дня возможности обмена информацией между агентом и Центром возможно $T_{БП} > T_{БП}^{max}$. Следовательно, резервирование БЗ у агента может не решить проблемы.

2 вариант. Размещение концептов, необходимых для работы агента у него и резервный способ передачи сообщений в Центр. Этот вариант представлен на рисунке 2.27.

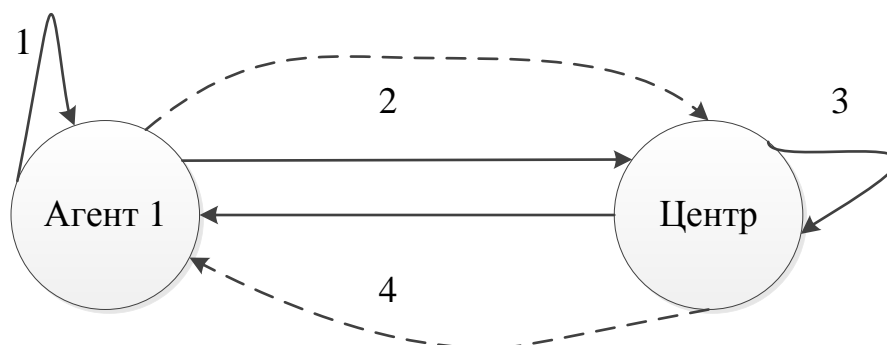


Рисунок 2.27. Схема БП в случае распределенной ИС, дублирование концептов у агентов и резервный способ передачи сообщений в Центр

В этом случае при сбоях в обмене информацией между Центром и Агентом 1 используется резервный способ доставки информации:

$$T_{БП} = t_{\text{обработки}}_{A1} + t_{\text{передачи}}_{A1 - \text{Центр}} + t_{\text{обработки}}_{\text{Центр}} + t_{\text{передачи}}_{\text{Центр} - A1}, \quad (2.23)$$

где

$t_{\text{обработки}}_{A1}$ – время работы агента1, включая человеко-машинное участие;

$t_{\text{передачи_рез}_{A1-\text{Центр}}}$ – время передачи результатов работы из агенту1 в Центр резервным способом;

$t_{\text{обработки}_{\text{Центр}}}$ – время работы Центра, включая человеко-машинное участие;

$t_{\text{передачи_рез}_{\text{Центр}-A1}}$ – время передачи результатов работы из Центра агенту1 резервным способом.

Следовательно, для рассматриваемой задачи необходимо резервировать базу знаний агента и разрабатывать резервный способ передачи данных.

Анализ архитектурных решений человеко-машинной распределенной системы до начала процесса кодирования позволяет снизить риск отказа такой системы в процессе эксплуатации.

В таблице 2.11 приведены соответствия элементов МППР элементам ИС [63].

Таблица 2.11

Соответствие элементов МППР элементам ИС

Элементы МППР	Элементы DFD	Элементы UML (диаграммы)			Элементы ИС		
		прецедент ов	классо в	последова тельности	ПИ	Код	Уровень БД
Ресурс	Поток данных, хранилище данных	нет	Класс	Класс, параметр метода	Поле ввода, таблица	Переменная, файловая переменная, таблица	Таблица
Преобразователь	Функция	Прецедент	Метод класса	Метод	Строка ввода, таблица	Функция	ХП
Агент	Внешняя сущность	Актер	Класс	Актер, класс	Интерфейс программного модуля	Код программного модуля	Таблицы, ХП

Таким образом, предложенный метод позволяет преобразовать КМПО МППР в КМПО ИС.

Следует отметить, что на сегодняшний момент предлагаются методы разработки ИС так или иначе затрагивающие анализ процессов ОТС. Кратко рассмотрим их.

Метод П. О. Скобелева

Метод Скобелева предназначен для создания МАС оперативной обработки информации для поддержки процессов принятия решений. В качестве модели ОТС предлагается модель сети потребностей и возможностей (ПВ-сеть) предприятия [72]. При этом каждое предприятие представляется в виде сети агентов потребностей и возможностей. Метод решает задачу взаимодействия этих агентов в процессе принятия решений.

Метод разработки П. О. Скобелева включает в себя следующие этапы: описание предметной области МАС; описание классов агентов и правил принятия решений; описание протоколов взаимодействия агентов; типов и структуры сообщений; программная реализация агентов.

Данный метод реализован в виде набор компонентов для разработки мультиагентных систем – Magenta Toolkit [73]. Настройка системы под конкретную предметную область обеспечивается с помощью инструмента для создания онтологий. Инструментарий предназначен для разработки МАС, связанных с планированием и распределением ресурсов. Он не занимается вопросами анализа и реинжиниринга БП.

Метод О. В. Карсаева и В. И. Городецкого

Метод О. В. Карсаева и В. И. Городецкого базируется на методологии Gaia [74] и среде MASDK [75], поддерживающей ее использование. Он предназначен для разработки прикладных многоагентных систем.

Метод состоит из десяти этапов, выполняемых в определенной последовательности [76]. Результаты каждого этапа определяют последовательность выполнения следующих этапов. Решения, описанные на одних этапах, являются исходными данными для выполнения последующих этапов. Укрупненно метод может быть описана в виде пяти стадий.

Первая стадия «Проектирование прикладной МАС» включает в себя два этапа:

- анализ предметной области;
- описание онтологии предметной области.

Результатом этапа анализа предметной области является задача идентификации классов агентов и сопоставления им ролей. Распределение ролей между классами агентов определяет, какие классы агентов на дальнейших этапах разработки будут обеспечивать решение определенных задач.

Вторая стадия «Проектирование классов агентов» занимается описанием трех компонент, образующих структуру агента:

- модель поведения агента;
- модели сервисов;
- ментальная модель.

На этой стадии идет только описание агентов с помощью диаграмм с использованием объектно-ориентированного подхода.

Написание программного кода происходит на третьей стадии. При этом описываются сервисные функции. После этого происходит автоматическая генерация программного кода классов агентов.

На четвертой стадии идет описание агента, состоящее из данных, необходимых для их установки, и знаний и правил его поведения.

На последней стадии происходит развертывание агентов в сети.

Таким образом, метод О. В. Карсаева и В. И. Городецкого не позволяет описать статические и динамические БП, а, следовательно, не занимается вопросами их анализа и реинжиниринга.

Метод А. Н. Швецова

Предложенный метод относится к разработке корпоративных интеллектуальных систем поддержки принятия решений [51]. На первоначальном этапе большое внимание уделяется описанию структурного, логического и поведенческого аспектов функционирования ОТС. На этапе формализации строятся структурно-логическая модель, база знаний, топологическая и объектная модели. Далее разрабатывается прототип системы и ее промышленный вариант. В данном методе основной упор делается на извлечение и формализацию знаний о предметной области.

Данный метод реализован в виде программного пакета DISIT (Distributed Intellectual System Integrated Toolkit) [77]. Он предназначен для разработки МАС, основан на следующих принципах:

- описание модели предметной области, с использованием фрейм-концептов;
- описание поведения агентов в виде продукций.

В данном инструментальном пакете последовательно выполняются следующие этапы метода:

- представление модели предметной области;
- наполнение модели логикой взаимоотношений фрейм-концептов и их атрибутов;
- выделение интеллектуальных агентов и определение их поведения с учетом системных ограничений;
- трансляция полученной концептуальной модели предметной области в структурно-логическую модель МАС;
- размещение интеллектуальных компонент и агентов в корпоративной сети.

Таким образом, метод А. Н. Швецова не позволяет описать статические и динамические БП, а, следовательно, не занимается вопросами их анализа и реинжиниринга.

Метод Д. В. Александрова

Александровым предложен метод моделирования распределенных систем управления БП предприятия [78]. При анализе предметной области предлагается использовать структурно функциональный подход. Имитационные модели разработаны на аппарате раскрашенных сетей Петри. По результатам имитационного моделирования предлагаются рекомендации по совершенствованию БП, а, при необходимости, проведение тактического реинжиниринга, заключающегося в добавлении\удалении функций, сотрудников, перераспределения функций между сотрудниками, перевод сотрудников из одного структурного подразделения в другое и т. п. Затем идет

реализация агентного приложения для автоматизации выполнения БП. Также метод предполагает использования ИМ для мониторинга БП предприятия.

Для сравнительной оценки методов разработки информационных систем предлагается следующий набор критериев:

- модель процессов ОТС, которая должна описывать статические и динамические БП, а также модель ЛПР;
- средства анализа процессов, включающие организационный реинжиниринг, анализ «узких мест»;
- возможность использования данных из модели ОТС при разработке ИС;
- использование структурного и объектно-ориентированного подходов;
- результаты автоматизации (бизнес-процессы, согласование решений, ППР – использование МЛВ).

В таблице 2.12 приведены результаты сравнения методов разработки ИС.

Таблица 2.12

Сравнение методов разработки ИС

Критерии сравнения	Метод Скобелева	Метод Городецкого, Карсаева	Метод Швецова	Метод Александрова	Новый Метод
1	2	3	4	5	6
1. Модель процесса ОТС	ПВ-сети	НЕТ	НЕТ	Раскрашенные сети Петри	Модель МППР
1.1. Статический бизнес-процесс	ДА	НЕТ	НЕТ	ДА	ДА
1.2. Динамический бизнес-процесс	НЕТ	НЕТ	НЕТ	НЕТ	ДА
1.3. Модель ЛПР	ДА	ДА	ДА	НЕТ	ДА
2. Средства анализа процессов					

Продолжение таблица 2.12

Критерии сравнения	Метод Скобелева	Метод Городецкого, Карсаева	Метод Швецова	Метод Александрова	Новый Метод
--------------------	-----------------	-----------------------------	---------------	--------------------	-------------

2.1. Реинжиниринг организационный	НЕТ	НЕТ	НЕТ	ДА	НЕТ
2.2. Анализ «узких мест»	НЕТ	НЕТ	НЕТ	НЕТ	ДА
3. Разработка ИС					
3.1. На основе модели ОТС в части динамических БП	НЕТ	НЕТ	НЕТ	НЕТ	ДА
3.2. На основе модели ОТС в части ЛПР	ДА	ДА	ДА	НЕТ	ДА
3.3. Структурный подход	НЕТ	НЕТ	НЕТ	ДА	ДА
3.4. ОО подход	ДА	ДА	ДА	НЕТ	ДА
4. Результаты автоматизации	ДА	ДА	ДА	ДА	ДА
4.1. Бизнес-процессы					
4.2. Согласование решений	ДА	НЕТ	ДА	НЕТ	ДА
4.2. ППР – использование МЛВ	НЕТ	НЕТ	ДА	НЕТ	ДА

Существующие методы не полностью решают задачу разработки ИС, затрагивающую анализ процессов ОТС. Они не учитывают динамику БП, недостаточно уделяют внимание анализу «узких» мест, не используют информацию из модели процессов ОТС в части динамических БП для разработки ИС. Предлагаемый метод решает эти задачи. Кроме того, метод уделяет внимание вопросу надежности человеко-машинной распределенной системы в условиях ограничений по срокам выполнения бизнес-процессов.

2.6. Анализ задержек в синхронной информационной системе

При проектировании ИС необходимо решить вопрос, на сколько децентрализованной должна быть ее структура, стоит ли отдельные

составляющие реализовывать в виде интеллектуальных агентов [79]. При этом встают вопросы о степени структуризации и выделении отдельных агентов, синхронизации работы и обмене между отдельными модулями ИС.

На первоначальном этапе проектирования модели МППР определяется некоторая исходная структура системы, производится выделение модулей, которые необходимо реализовать в виде программных агентов. Однако, на этапе описания БП более детальная структуризация не всегда очевидна. Выделять или нет отдельные модули или оставить один программный модуль – это понимание приходит уже позже, иногда, только после пробной эксплуатации ИС. Для своевременного решения этих вопросов целесообразно использовать имитационное моделирование.

Как правило, требуемые для нормального функционирования готовой ИС объемы и характер информационного обмена между отдельными компонентами известны или задаются. Практически информационный обмен между удаленными компонентами системы сможет быть реализован в различных протоколах и сетевых решениях. Моделирование взаимодействия потенциальных подсистем с использованием реалистических параметров позволяет проектировщику принять решение о необходимости автономизации подсистем в программные модули.

В качестве примера такого моделирования рассмотрим обмен, связанный с заходом клиента с удаленного компьютера на сервер и работы с графическим интерфейсом пользователя. Оценки потребного ресурса пропускной способности канала по средним характеристикам могут оказаться существенно заниженными. В результате это может приводить к непредвиденным задержкам в обслуживании, что периодически и наблюдается в аналогичных ИС, построенных по традиционной методике.

Пусть «заявкой» будет обращение клиента (оператора) к системе посредством GUI через интернет-канал. «Обработка» же запроса заключается в заполнении оператором некоторой формы. В этом случае время обработки запроса будет измеряться минутами. Альтернативой может быть создание двух

программных агентов, работающих независимо и параллельно, а обмен осуществляется посредством почтового клиента.

Интенсивность запросов в реально существующих ИС с удаленным доступом существенно стохастический и нестационарный процесс. Приводим на рис. 2.28 зависимость частоты запросов от времени в реальной ИС.

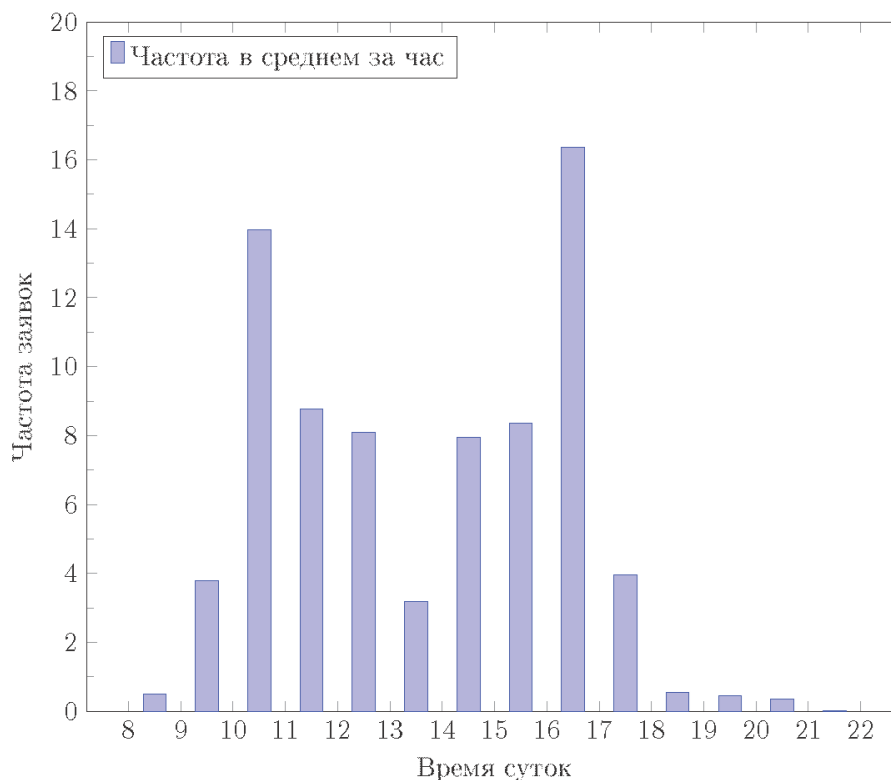


Рисунок 2.28. Распределение заявок в течение суток

Как видим с одной стороны в некоторые периоды времени частота запросов существенно возрастает, по сравнению со среднесуточной. Однако экономически вряд ли целесообразна избыточность ресурсов, ориентированная на максимальные показатели.

Имитационное моделирование осуществлялось в простейшей модели системы массового обслуживания (СМО) $m/m/1$ [80], поскольку полученных результатов достаточно для принципиального решения о разбиении на подсистемы. Еще в работе А. Дорохова (CERN Bulletin dec, 1999 [81]) было показано, что результаты численного моделирования $m/m/1$ (и ряда других) прекрасно согласуются со значениями тех числовых характеристик СМО, которые можно получить в рамках простой теории.

Потоки заявок и обслуживания рассматриваем как пуассоновские с интенсивностями λ и μ , соответственно. При этом промежутки времени между последовательными заявками T_a и время обслуживания T_s имеют показательное распределение с соответствующими параметрами. Существенным оказывается важное свойство данного распределения, а именно – математическое ожидание и среднеквадратичное отклонение равны $1/\lambda$ ($1/\mu$). Примерно $2/3$ значений случайной величины, имеющей показательное распределение, меньше математического ожидания и $1/3$ больше (рис. 2.29).

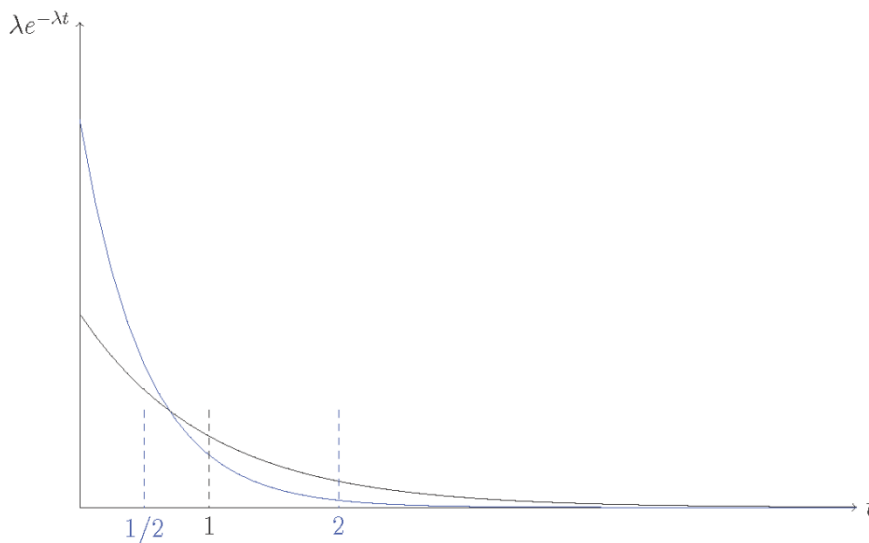


Рисунок 2.29. Плотности вероятности показательного распределения с $\lambda = 1$ и $\lambda = 2$

Это приводит к интересному эффекту. Пусть $T_a = 2T_s$, иначе говоря никакой очереди при регулярном периодическом поступлении заявок не может быть. Более того, половину времени СМО простаивает в состоянии ожидания.

«Хвосты» на кривых распределения приводят к тому, что значительное количество заявок приходит чаще, чем через T_a , а время обслуживания превышает T_s . Насколько значительным будет увеличение среднего времени обслуживания можно судить по значению мгновенной очереди q_c .

Алгоритм расчета следующий. Последовательно с помощью встроенного генератора случайных чисел получаем значения времени прихода и времени обслуживания (с соответствующими параметрами). Сравнивая эти времена, надо либо увеличить на единицу, либо уменьшить количество заявок в очереди.

Указанный алгоритм был реализован в языке свободно распространяемой программы *R*, использующей *Mersenne-Twister* алгоритм. Кроме того, был добавлен блок визуализации полученных результатов в виде скрипта для программы *TikZ*, позволяющий получить графики в виде *pdf* файлов.

Полученные значения мгновенной длины очереди (на рис. 2.30 доходит до 4) дают проектировщику основание для оптимальной структуризации системы и введения программных агентов. Аналогичное моделирование проводится для всех «узких» мест БП. Окончательное решение о структуризации модуля принимается исходя из результатов моделирования МППР.

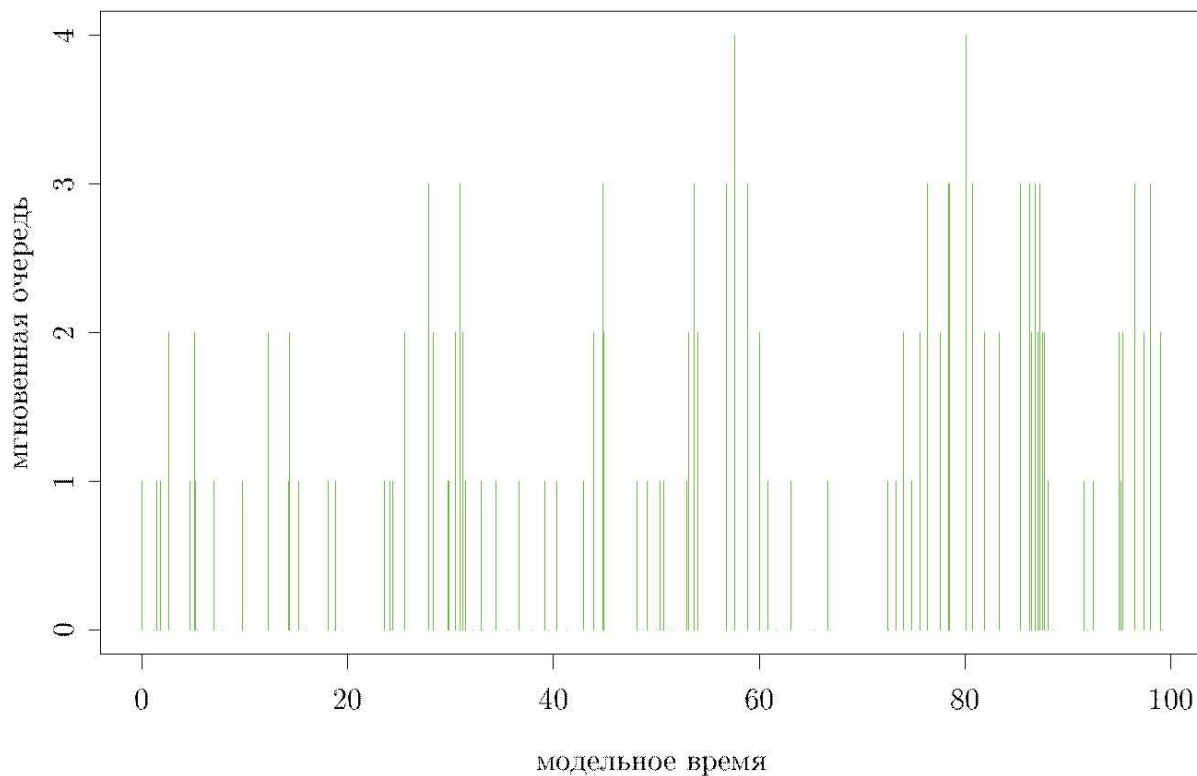


Рисунок 2.30. Длина мгновенной очереди при $T_s/T_a=4/3$

Использование имитационного моделирования для анализа работы отдельных элементов ИС позволяет оценить предлагаемые решения по архитектуре ПО до момента его эксплуатации, а, следовательно, имеется возможность их скорректировать в процессе разработки.

2.7. Методика оценки эффективности работы метода разработки информационных систем

Оценим эффективность работы предложенного метода проектирования ПО. Одним из главных критериев оценки является время выполнения проекта по созданию ПО – $T_{ПО}$. Оно складывается из времени выполнения каждого этапа проекта и времени перехода между этапами. На практике на разных этапах разработки участвуют разные специалисты, поэтому требуется адаптация результатов предыдущего этапа для специалистов следующего, поэтому и возникает время перехода между этапами.

$$T_{ПО} = T_{обсл.} + T_{форм.} + T_{модел.} + T_{проект.} + T_{разр.} + T_{переход.} \quad (2.24)$$

где $T_{обсл.}$ – время проведения обследования (1 этап);

$T_{форм.}$ – время формализации БП, т. е. время построения модели (2 этап);

$T_{модел.}$ – время проведения моделирования и реинжиниринга (3 этап);

$T_{проект.}$ – время проектирования ПО (4 этап);

$T_{разр.}$ – время разработки ПО (5 этап);

$T_{переход.}$ – суммарное время, затрачиваемое при переходе от одного этапа проекта к другому.

$$T_{переход.} = T_{обсл. \rightarrow форм.}^{переход.} + T_{форм. \rightarrow модел.}^{переход.} + T_{модел. \rightarrow проект.}^{переход.} + T_{проект. \rightarrow разр.}^{переход.} \quad (2.25)$$

где $T_{i \rightarrow j}^{переход.}$ – время перехода между i и j этапами.

Время проведения обследования $T_{обсл.}$ и время перехода между 1 и 2 этапами $T_{обсл. \rightarrow форм.}^{переход.}$ не зависит от того, использовался или нет предлагаемый метод.

В процессе разработки ИС строятся DFD-диаграммы и диаграммы языка UML. Следовательно время проектирования складывается из двух составляющих:

$$T_{проект.} = T_{DFD} + T_{UML} + T_{DFD \rightarrow UML}^{переход.} \quad (2.26)$$

где $T_{DFD.}$ – время построения всех DFD-диаграмм;

T_{UML} – время построения модели архитектуры ПО в виде UML-диаграмм;

$T^{DFD \rightarrow UML}_{переход.}$ – время перехода от построения DFD-диаграмм к UML-диаграммам. (На практике связано с передачей знаний от аналитика БП к архитектору ПО.)

Время построения одной DFD-диаграммы определяется количеством функциональных блоков и потоков данных. Следовательно,

$$T_{DFD.} = \sum(n * t_{бл.} + m * t_{поток}), \quad (2.27)$$

где n – количество функциональных блоков;

$t_{бл.}$ – время рисования одного блока;

m – количество потоков данных;

$t_{поток}$ – время рисования одного потока.

Предложенный метод использует следующие три вида UML-диаграмм: прецедентов, последовательности и классов. Значит,

$$T_{UML} = T_{UC} + T_{Int} + T_{Class}, \quad (2.28)$$

где T_{UC} – время построения всех диаграмм прецедентов;

T_{Int} – время построения всех диаграмм последовательности;

T_{Class} – время построения всех диаграмм классов.

Время построения одной диаграммы прецедентов, в общем случае, зависит от количества вариантов использования ПО. При ее автоматической генерации на основе данных из DFD-диаграммы эта зависимость незначительна по сравнению с ручным моделированием. Следовательно, при использовании предложенного метода время построения одной диаграммы прецедентов можно считать величиной постоянной (t_{UC}), тогда

$$T_{UC.} = \sum t_{UC}. \quad (2.29)$$

При использовании предлагаемого метода $T^{DFD \rightarrow UML}_{переход.}$ будет существенно меньше за счет процесса автоматизации моделирования. В случае использования метода снижается человеческий фактор потери части информации при переходе от одной модели к другой.

В предлагаемом методе предлагается автоматически генерировать заготовки программных модулей, описывающих классы и формы ПИ. Следовательно, время, затраченное на переход от этапа проектирования к этапу

разработки ПО $T_{\text{проект} \rightarrow \text{разр. переход.}}$ сокращается по сравнению с другими методами.

Анализ предложенного метода показывает, что он позволяет уменьшить следующие времена: $T_{DFD \rightarrow UML_{\text{переход.}}}$, $T_{UC.}$, $T_{\text{проект} \rightarrow \text{разр. переход.}}$.

Вывод по главе 2:

Во второй главе было проведено исследование предметной области МППР и описаны основные объекты КМПО МППР, а также предметной области ИС и построена КМПО ИС. Разработаны модель и метод в области создания ИС, которые отличаются от существующих:

- 1) методикой системного анализа и моделью для формализации МППР. При этом учитывается наличие ЛПР, которые могут быть представлены в виде ИА;
- 2) ИМ для проверки модели «как будет» на этапе реинжиниринга БП, и оценки производительности ИС;
- 3) интеллектуальностью разработки ИС, включающей функциональный и объектно-ориентированный анализ, моделирование ПИ, формирование исполняемого кода ИС;

Анализ надежности архитектурных решений при автоматизации процессов ОТС в условиях ограничений по срокам выполнения БП особенно эффективен в случае географически распределенной системы.

Предложенный метод ППР разработки ИС мультиагентных процессов преобразования ресурсов позволяет уменьшить время, затрачиваемое на переход от функционального моделирования к объектно-ориентированному, а также от проектирования к разработке ПО.

3. CASE-средство Bpsim.SD

3.1. Функциональные возможности пакета Bpsim.SD

Bpsim.SD представляет собой CASE-средство автоматизации процесса проектирования DFD, IDEF0 диаграмм, UML диаграмм прецедентов, последовательности и классов, а также ПИ разрабатываемой ИС [55; 82-84].

В рамках процесса моделирования архитектуры ИС Bpsim.SD предлагает пользователю следующие возможности:

1. Описание бизнес-процессов, автоматизируемых разрабатываемой ИС, с помощью диаграмм стандарта DFD. Диаграммы DFD декомпозируемы до любого уровня детализации. Автоматическое создание DFD-диаграмм на основе модели МППР.

2. Описание функций, выполняемых пользователями системы в рамках автоматизируемых процессов, с помощью диаграмм прецедентов. Допускается построение диаграмм прецедентов с «нуля», т. е. создание новой диаграммы, или с помощью автоматизированного конвертирования из DFD, где выбранным из списка существующих на диаграмме DFD внешним сущностям ставятся в соответствие актеры, а функциям – прецеденты. Полученные конвертированием диаграммы редактируемы. Допускается строить неограниченное количество диаграмм прецедентов для каждой диаграммы DFD.

3. Для каждого прецедента, выполняемого пользователем, может быть дано описание последовательности элементарных операций в системе. Для этого предназначены диаграммы последовательности, которые могут быть созданы для любой функции диаграммы прецедентов. Система позволяет частично автоматизировать этот процесс: разработчику предлагается перенести на диаграмму последовательности выбранных актеров с диаграммы прецедентов. Для каждого объекта диаграммы последовательности определяется один из четырех типов по принадлежности к определенному пакету (границы, актеры, управление, бизнес-объекты).

4. Создание диаграммы классов и сопоставление объектов диаграммы последовательности (кроме границ) с классами этой диаграммы.

5. Проектирование визуальных форм моделируемого ПО (границ диаграммы последовательности): размещение на форме компонентов, привязка к компонентам методов и свойств классов и сохранение моделей форм в форматах .dfm и .pas. В дальнейшем данные модели передаются программисту для импорта в среду Delphi и дальнейшей проработки алгоритмов.

6. Формирование отчетов о созданном проекте с изображениями спроектированных диаграмм и форм. Предусмотрен вывод отчетов на печать и экспорт в Word.

7. Сохранение проекта на сервере и загрузка с сервера MS SQL Server для редактирования.

3.2. Описание CASE-средства Bpsim.SD

3.2.1. Общая структура CASE-средства BPsim.SD

CASE-средство Bpsim.SD (SD) состоит из трех подсистем: создание DFD-диаграмм, UML-диаграмм, моделирование ПИ. Структура Bpsim.SD представлена на рисунке 3.1 [85].

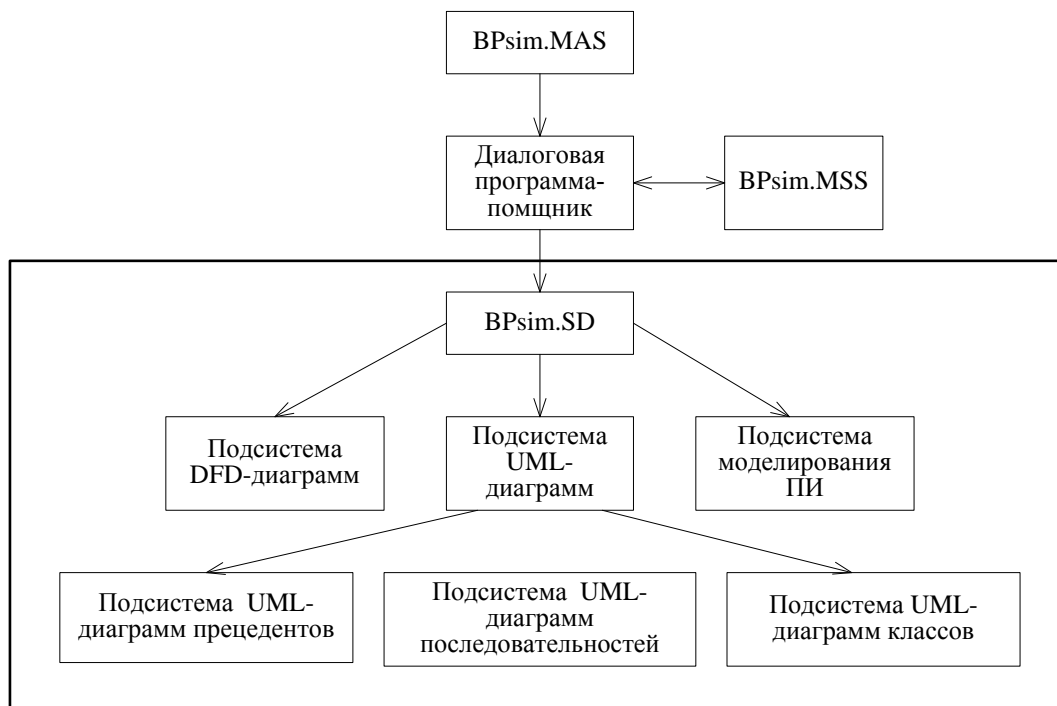


Рисунок 3.1. Структура CASE-средства Bpsim.SD

Более подробно подсистемы рассмотрены в следующих разделах.

После ввода идентификационных данных пользователя и установления соединения с сервером открывается главная форма системы, представленная на рисунке 3.2.

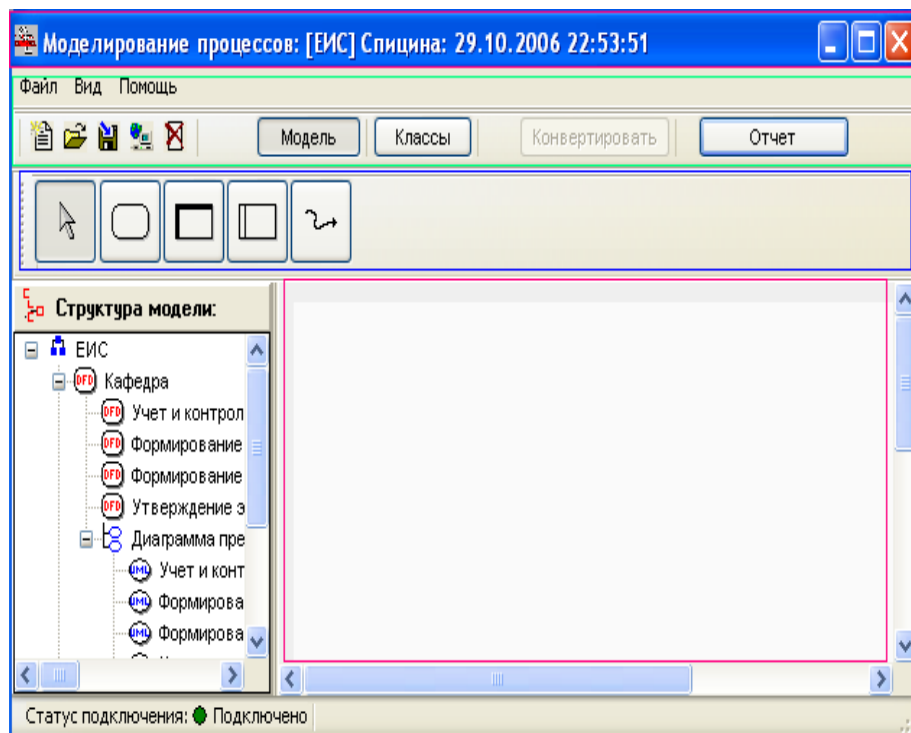


Рисунок 3.2. Главная форма Bpsim.SD

Главная форма состоит из следующих частей:

- заголовок, отражающий информацию о текущем проекте (название, авторы, дата и время создания);
- главное меню системы предоставляет пользователю доступ ко всем функциям приложения;
- панели инструментов – дублируют основные команды главного меню, на ней располагаются инструменты создания и редактирования диаграмм классов;
- область создания диаграммы – здесь происходит непосредственное создание и редактирование диаграмм классов;
- строка состояния – отображает информацию о текущем состоянии подключения к выбранному SQL Server.

3.2.2. Создание диаграммы DFD

После создания нового проекта в области панели инструментов автоматически активизируются кнопки с объектами диаграммы DFD (рисунок 3.3).

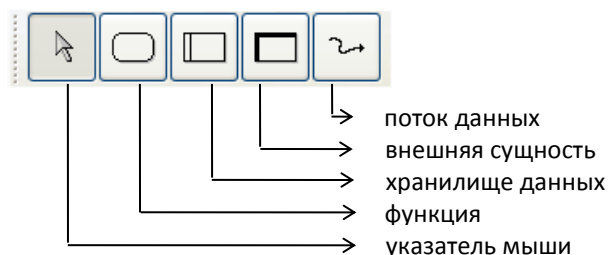


Рисунок 3.3. Панель инструментов диаграммы DFD

Для размещения объекта в области диаграммы необходимо щелкнуть на соответствующей кнопке панели инструментов и в том месте диаграммы, куда требуется поместить объект.

При наведении мышью над объектом или щелчке на нем, он выделяется красным цветом. За правую и нижнюю границы, а также за правый нижний угол объекта можно изменить его размер, удерживая нажатой левую клавишу мыши (рисунок 3.4).



Рисунок 3.4. Изменение размеров объекта

Для соединения объектов потоками данных щелкните на соответствующей кнопке панели инструментов. При наведении указателя мыши объекты (в том числе границы формы), которые можно соединять, выделяются синим цветом. Над границами формы и объектов курсор также меняет свой вид, давая понять, что отсюда (сюда) можно провести поток

(рисунок 3.5). Соединяемые в данный момент объекты также выделяются синим цветом.



Рисунок 3.5. Соединение объектов потоками данных

В контекстном меню «Тип» потока данных можно выбрать один из 5 типов (рисунок 3.6).

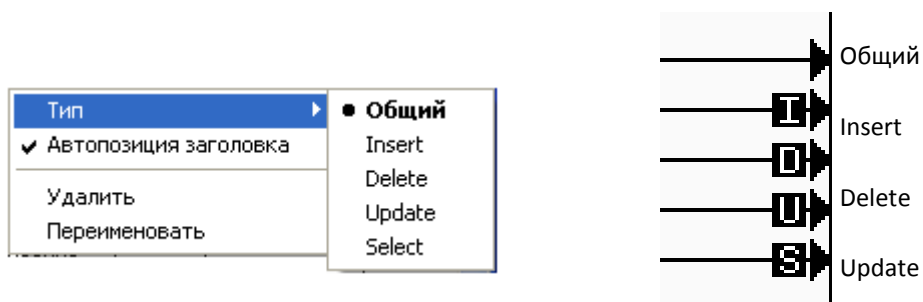
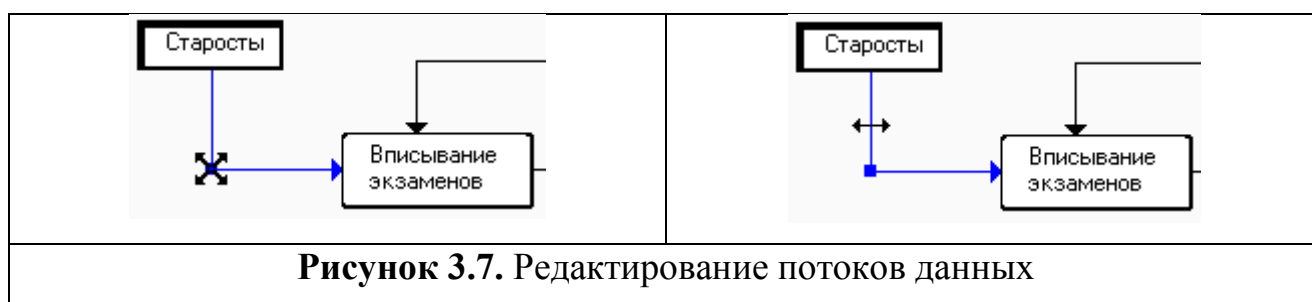


Рисунок 3.6. Типы потоков данных

Перемещение объектов и заголовков потоков осуществляется с нажатой левой клавишей мыши, при этом все потоки данных, связанные с данным объектом, автоматически перестраиваются. Редактирование потоков данных осуществляется перетаскиванием узлов или отрезков линий (рисунок 3.7). Автоматическое размещение заголовка потока данных вызывается через пункт «Автопозиция заголовка» контекстного меню потока (рисунок 3.6).



Все размещенные на диаграмме DFD функции заносятся в дерево проекта. Переименование объектов вызывается из контекстного меню объекта или соответствующего элемента дерева (рисунок 3.8).

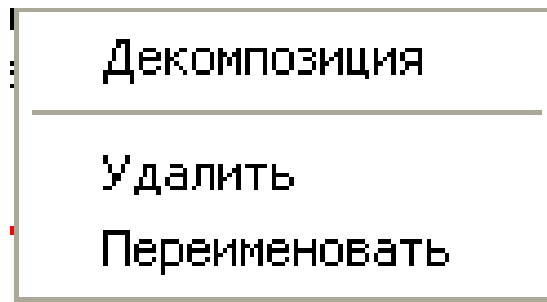


Рисунок 3.8. Контекстное меню функций диаграммы DFD

Для потоков данных открывается дополнительное окно (рисунок 3.9), позволяющее выбрать имя из списка уже существующих в данном проекте.

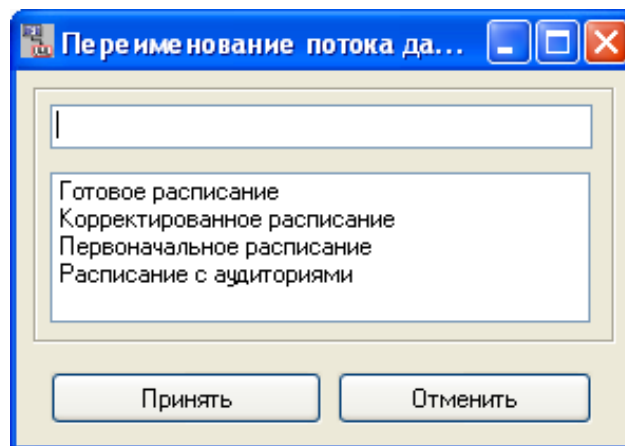


Рисунок 3.9. Форма «Переименование потока данных»

Аналогичные окна открываются для хранилищ данных и внешних сущностей при выборе пункта «Выбрать из существующих» контекстного меню (рисунок 3.10).

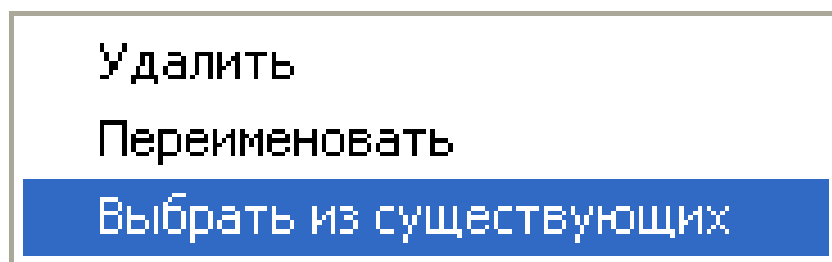


Рисунок 3.10. Контекстное меню хранилищ данных
и внешних сущностей диаграмм DFD

Для удаления объектов выберите пункт **«Удалить»** контекстного меню объекта или элемента дерева проектов (рисунок 3.8). Все связанные с данным объектом потоки также будут удалены.

Для создания декомпозиции диаграммы необходимо в контекстном меню декомпозируемой функции выбрать пункт **«Декомпозиция»** (рисунок 3.8). Затем в открывшемся окне необходимо указать первоначальное количество элементов (функций) в декомпозиции (рисунок 3.11).

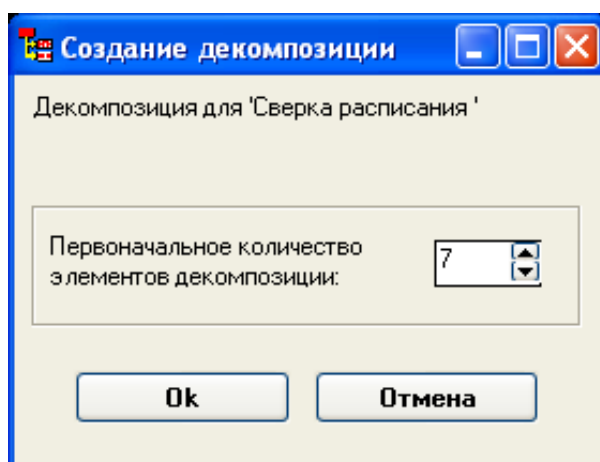


Рисунок 3.11. Форма «Создание декомпозиции»

После нажатия **«ОК»** автоматически создается новая диаграмма, содержащая указанное количество функций. В дерево проектов эти функции помещаются в качестве потомков элемента – декомпозируемой функции.

На рисунке 3.12 приведен пример готовой диаграммы DFD.

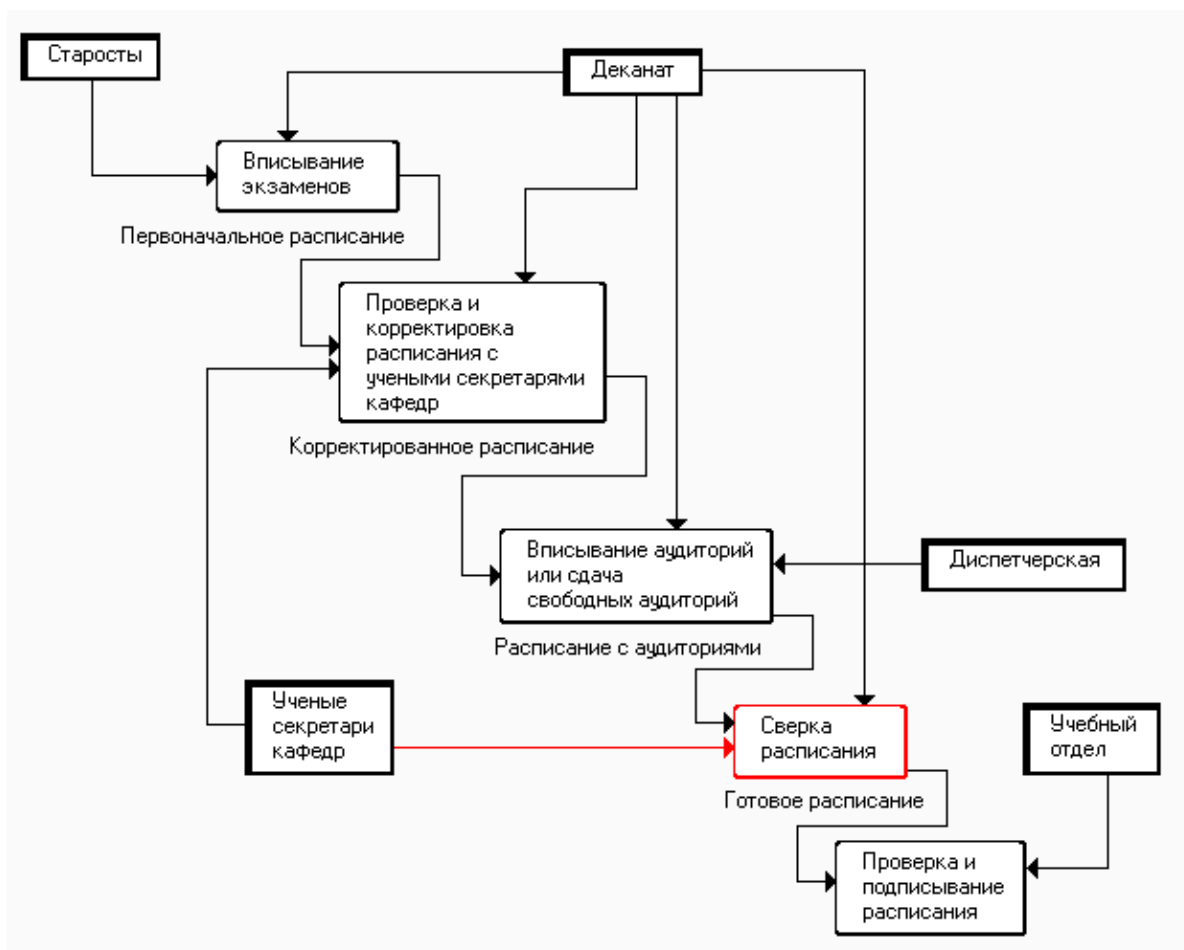


Рисунок 3.12. Диаграмма DFD

3.2.3. Создание диаграммы прецедентов

Создание диаграммы прецедентов осуществляется конвертированием из диаграммы DFD в соответствии с разработанной методикой. При этом функции диаграммы DFD заменяются на функции диаграммы прецедентов, а внешние сущности – на актеров.

Для запуска процесса конвертирования необходимо нажать кнопку «Конвертировать» на панели инструментов (рисунок 3.13).

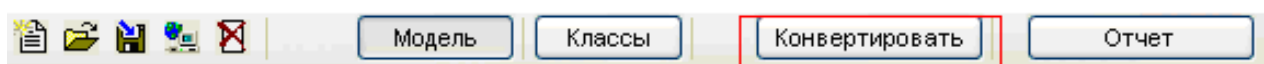


Рисунок 3.13. Стандартная панель инструментов

В появившемся окне (рисунок 3.14) выбираются существующие на данной диаграмме функции и внешние сущности, которые необходимо отобразить на диаграмме прецедентов. При необходимости сохранения связей между этими объектами устанавливается соответствующий флаг внизу окна.

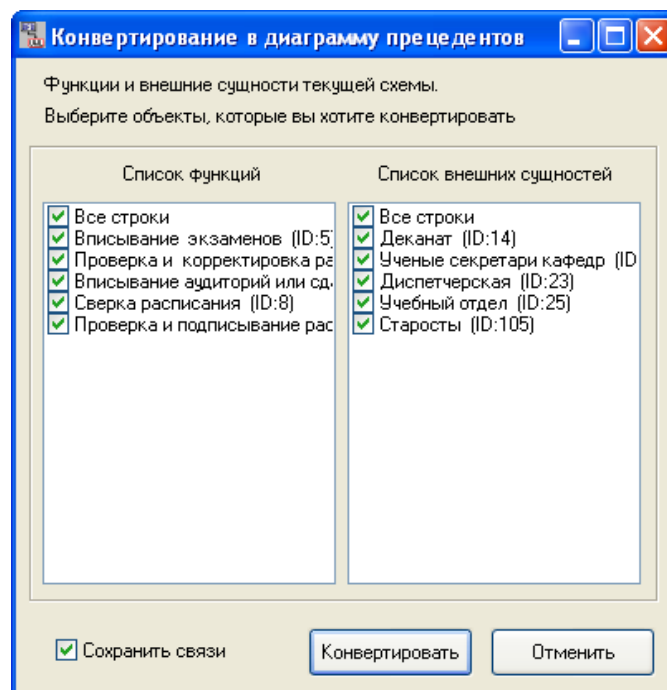


Рисунок 3.14. Форма «Конвертирование в диаграмму прецедентов»

В результате конвертирования в дереве проектов появляется соответствующий узел со списком существующих на диаграмме прецедентов функций. Построенная диаграмма приведена на рисунке 3.15.

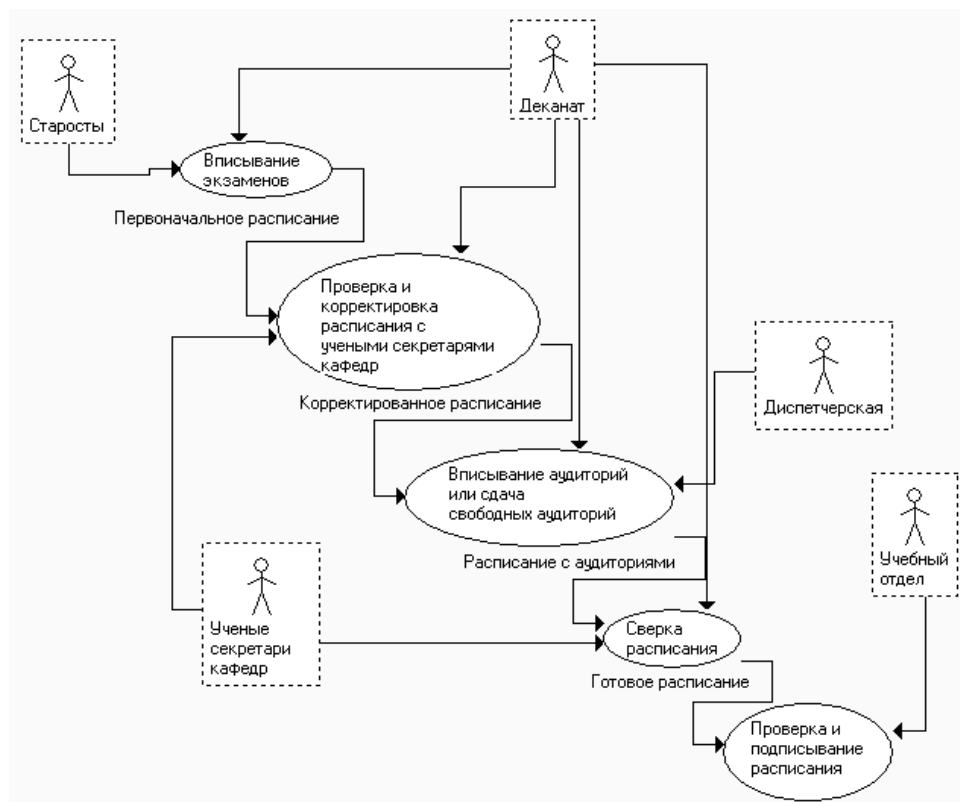


Рисунок 3.15. Диаграмма прецедентов

Разрешается создавать новую или редактировать существующую диаграмму с помощью панели инструментов диаграммы прецедентов (рисунок 3.16).

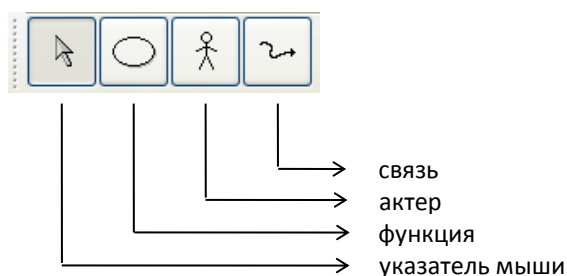


Рисунок 3.16. Панель инструментов диаграммы прецедентов

Редактирование диаграммы прецедентов осуществляется аналогично редактированию диаграмм DFD, и было описано выше.

3.2.4. Создание диаграммы последовательности

Диаграмма последовательности может быть создана для любой функции диаграммы прецедентов из ее контекстного меню (рисунок 3.17).

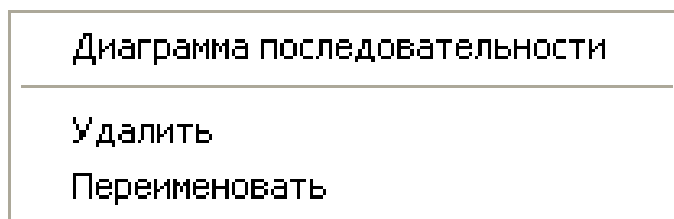


Рисунок 3.17. Контекстное меню функции диаграммы прецедентов

В открывшемся окне (рисунок 3.18) необходимо выбрать из списка существующих актеров, которые будут помещены на диаграмму последовательности.

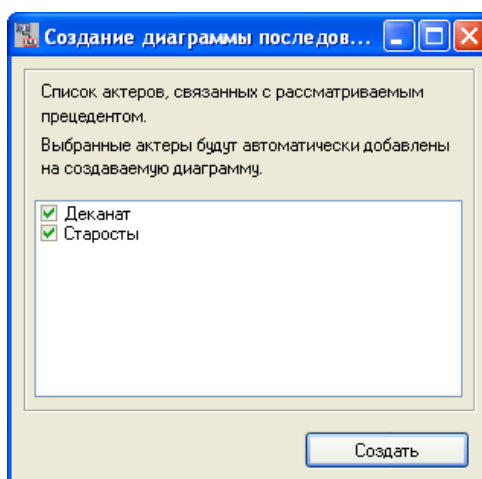


Рисунок 3.18. Форма «Создание диаграммы последовательности»

После создания диаграммы соответствующий узел добавляется в дерево проектов и активизируется панель инструментов для редактирования данной диаграммы (рисунок 3.19).

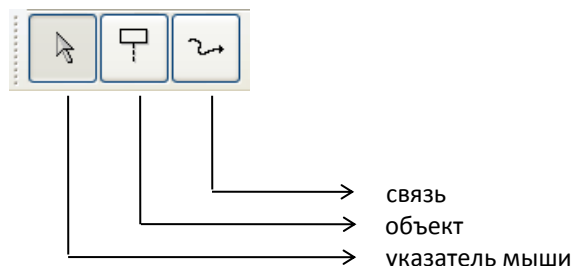


Рисунок 3.19. Панель инструментов диаграммы последовательности

Для добавления на диаграмму объекта достаточно щелкнуть на соответствующей кнопке, расположенной на панели инструментов, при этом объект помещается на форму автоматически следом за предыдущим объектом и имеет неопределенный тип (рисунок 3.20).



Рисунок 3.20. Типы объектов диаграммы последовательности

Назначение объектам диаграммы конкретных типов осуществляется через контекстное меню (рисунок 3.21).

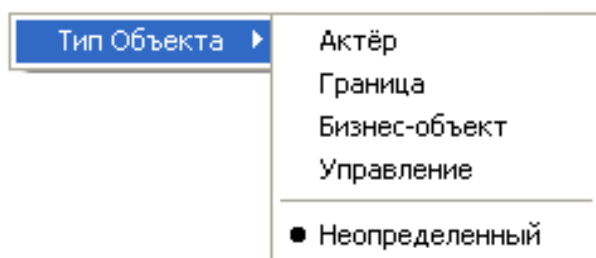


Рисунок 3.21. Контекстное меню неопределенного объекта диаграммы последовательности

Пользователю разрешается менять объекты местами (перетаскивание с нажатой левой клавишей мыши) и изменять их размеры. При изменении высоты какого-либо объекта, меняется высота всех остальных объектов.

Чтобы связать между собой два объекта, необходимо провести связи между осями времени данных объектов. Это осуществляется аналогично соединению объектов связями в диаграммах DFD и прецедентов, но в отличие от них, в диаграммах последовательности связь может начинаться и заканчиваться на одном и том же объекте (рисунок 3.22).

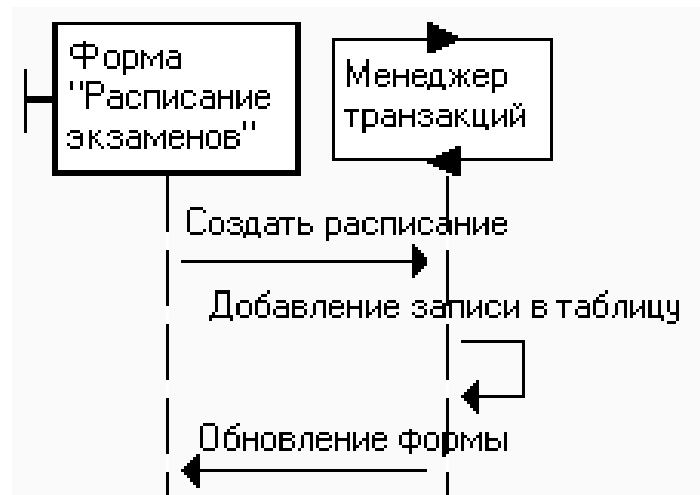


Рисунок 3.22. Диаграмма последовательности

Некоторым объектам диаграммы последовательности (актерам, бизнес-объектам и объектам управления) можно ставить в соответствие классы, созданные в данном проекте. Это осуществляется через пункт «Класс» контекстного меню (рисунок 3.23).

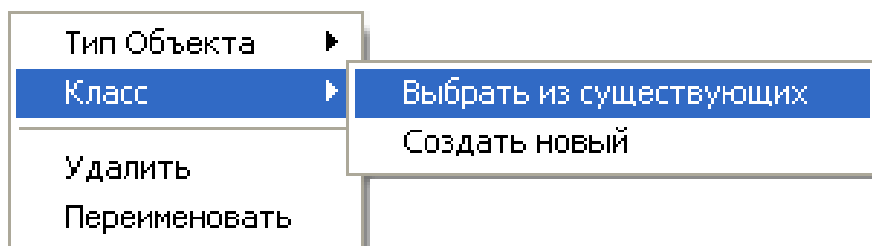


Рисунок 3.23. Контекстное меню объекта диаграммы последовательности

Если щелкнуть мышью на пункте «Выбрать из существующих», то открывается окно (рисунок 3.24), содержащее список всех созданных в проекте классов с описанием их свойств и методов.

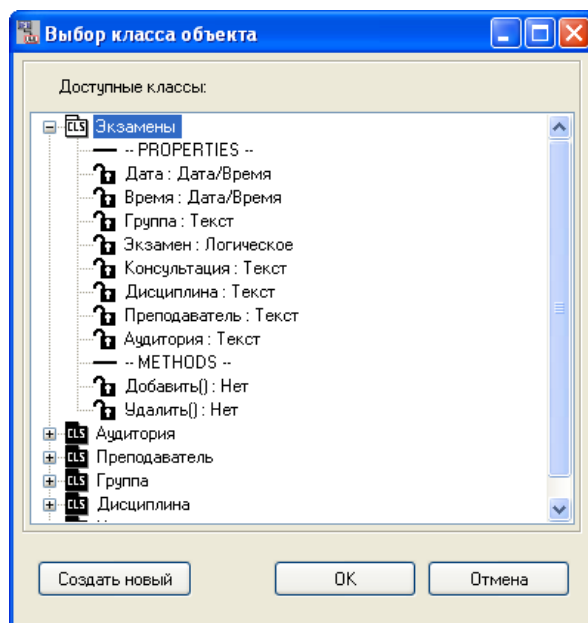


Рисунок 3.24. Форма «Выбор класса объекта»

При выборе пункта «Создать новый» в контекстном меню (рисунок 3.23) и в данном окне (рисунок 3.24) осуществляется переход на диаграмму классов для ее редактирования. Назначенный объекту класс прописывается в его заголовке на диаграмме последовательности (рисунок 3.25).



Рисунок 3.25. Бизнес-объект диаграммы последовательности

Пример диаграммы последовательности для прецедента «Составление расписания экзаменов» приведен на рисунке 3.26.

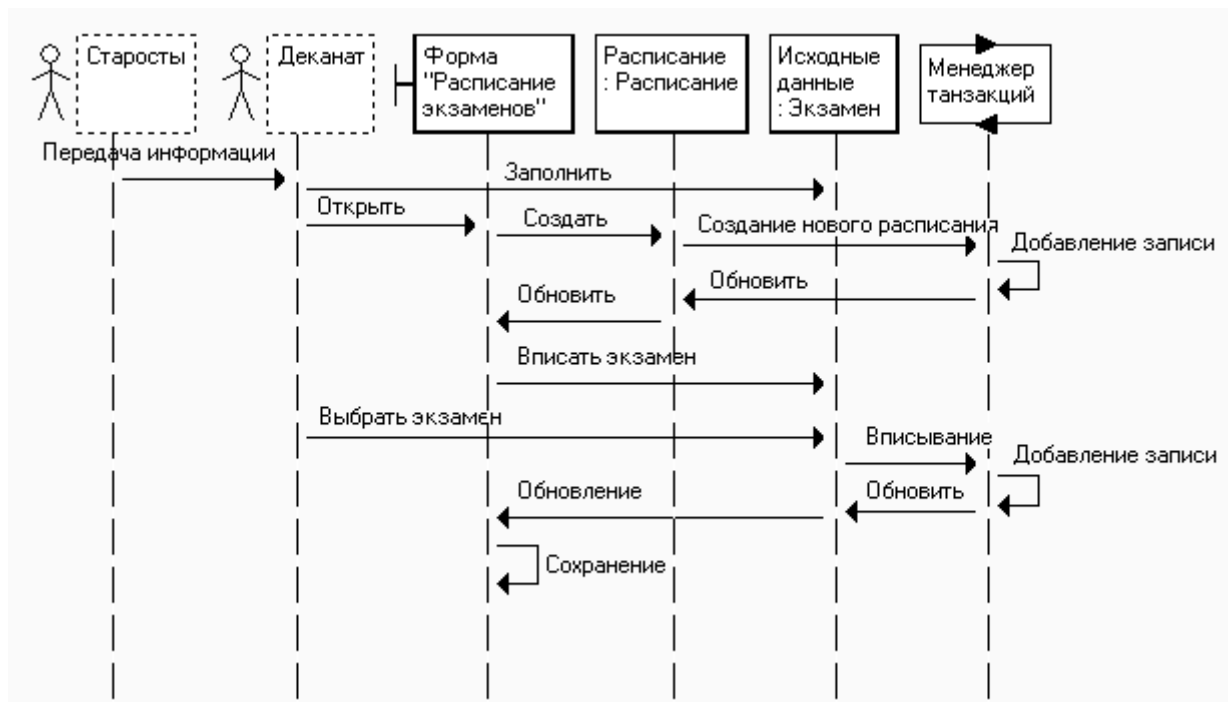


Рисунок 3.26. Диаграмма последовательности

Сохранение проекта на сервере осуществляется через пункт главного меню **«Сохранить проект»** или с помощью кнопки на стандартной панели инструментов (см. рисунок 3.13).

3.2.5. Создание диаграммы классов

Для перехода на диаграмму классов необходимо выбрать соответствующий пункт главного меню или воспользоваться панелью инструментов. Редактирование диаграммы классов выполняется с помощью панели инструментов, приведенной на рисунке 3.27.

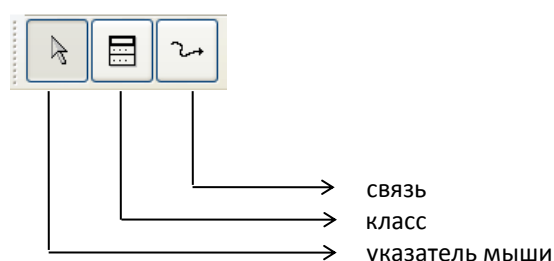


Рисунок 3.27. Панель инструментов диаграммы классов

Для размещения объекта в области диаграммы необходимо щелкнуть на соответствующей кнопке панели инструментов и в том месте диаграммы, куда требуется поместить объект.

При наведении мышью над объектом или щелчке на нем, он выделяется красным цветом. За правую и нижнюю границы, а также за правый нижний угол объекта можно изменить его размер, удерживая нажатой левую клавишу мыши. Аналогичным образом осуществляется перемещение границы, разделяющей свойства и методы редактируемого класса (рисунок 3.28).



Рисунок 3.28. Изменение размеров объекта

При выборе пункта «Авторамер» контекстного меню (рисунок 3.29) размер класса и положение разделяющей границы устанавливаются автоматически так, чтобы были видны все свойства и методы.

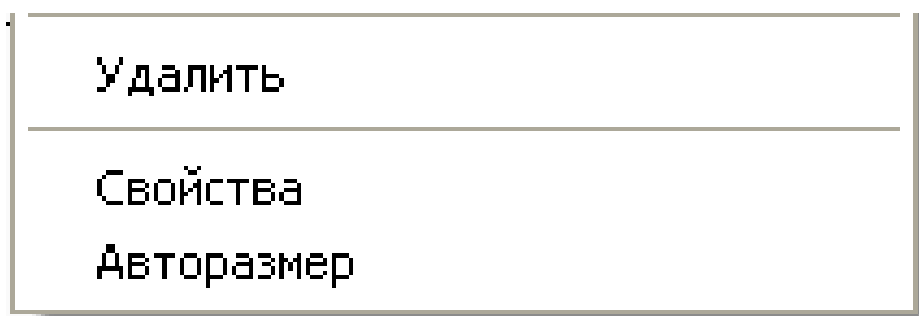
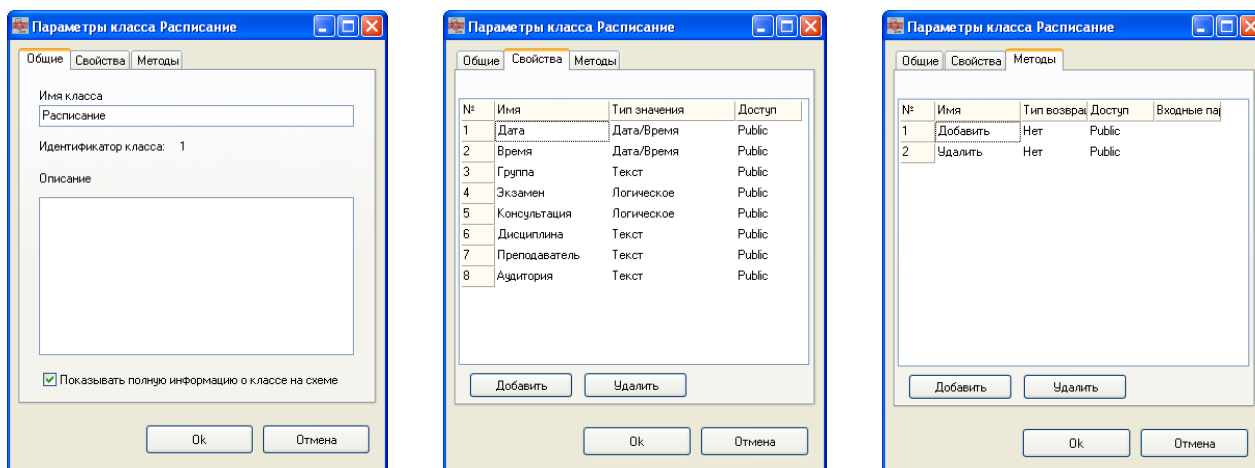


Рисунок 3.29. Контекстное меню класса

Двойной щелчок на созданном классе или выбор пункта «Свойства» контекстного меню открывают окно для ввода информации об этом классе (рисунок 3.30). Окно содержит 3 вкладки:

- вкладка «Общие»: задаются имя и описание класса;
- вкладка «Свойства»: задаются свойства класса с указанием имени, типа данных и вида доступа;
- вкладка «Методы»: задаются методы класса с указанием имени, типа возвращаемого значения, вида доступа и входных параметров.



«Общие»

«Свойства»

«Методы»

Рисунок 3.30. Форма «Параметры класса»

Для задания списка входных параметров необходимо дважды щелкнуть мышью на соответствующем поле вкладки «Методы» и после заполнения таблицы входных параметров нажать «Принять».

Для соединения классов связями щелкните на соответствующей кнопке панели инструментов. При наведении указателя мыши классы (в том числе границы формы), которые можно соединять, выделяются синим цветом. Над границами форм и классов курсор также меняет свой вид, давая понять, что отсюда (сюда) можно провести поток (рисунок 3.31). Соединяемые в данный момент классы также выделяются синим цветом.

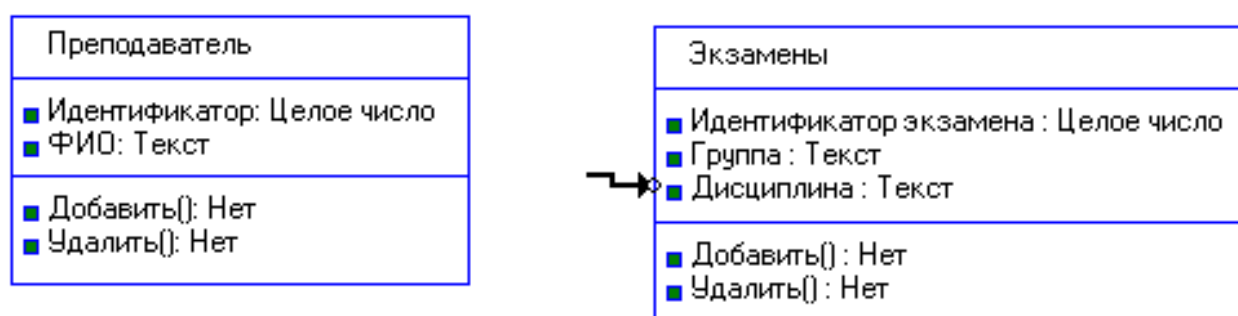


Рисунок 3.31. Соединение классов связями

Перемещение объектов – классов и заголовков связей осуществляется с нажатой левой клавишей мыши, при этом все связи, соединенные с данным объектом, автоматически перестраиваются.

Редактирование связей осуществляется перетаскиванием узлов или отрезков линий (рисунок 3.32).

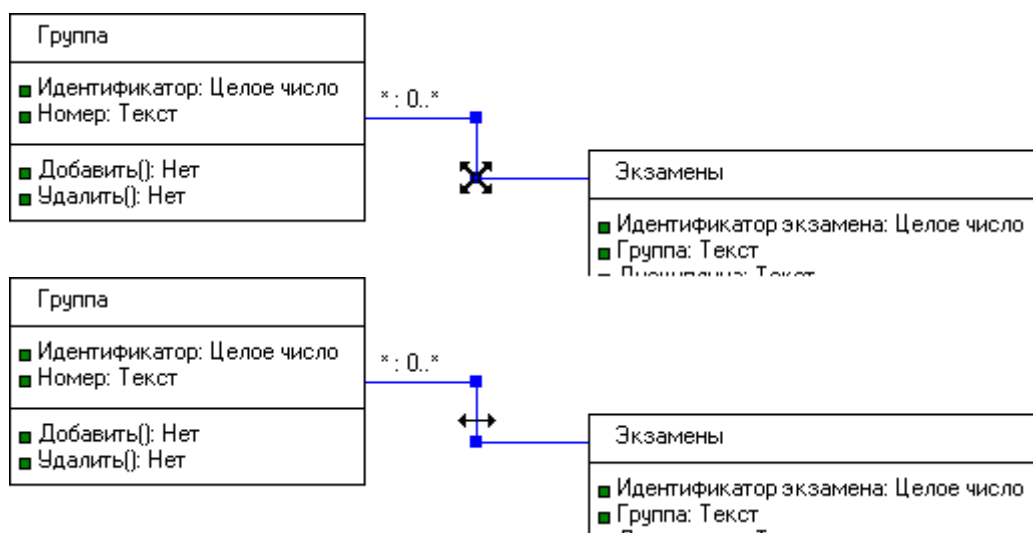


Рисунок 3.32. Редактирование потоков данных

Переименование классов осуществляется в служебном окне «Свойства» (рисунок 3.30) на вкладке «Общие», вызываемом по двойному щелчку на классе или через контекстное меню (рисунок 3.29).

При переименовании связей открывается дополнительное окно (рисунок 3.33), позволяющее выбрать имя из списка уже существующих в данном проекте потоков данных и связей.

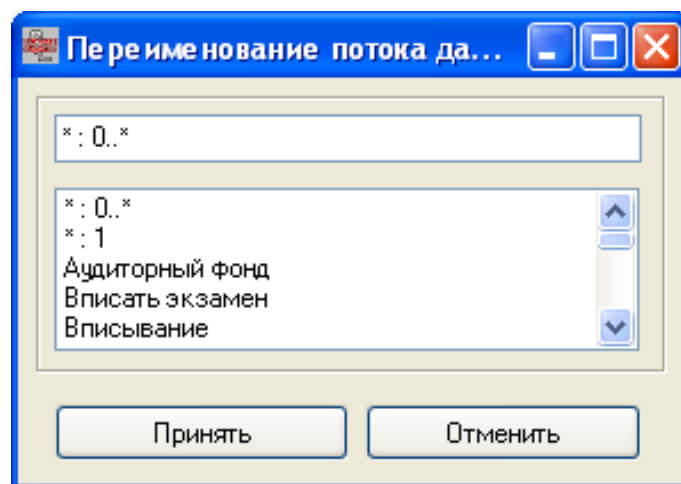


Рисунок 3.33. Форма «Переименование потока данных»

Для отображения полной информации о классе на диаграмме необходимо установить соответствующий флажок в параметрах класса на вкладке «Общие».

Для удаления объектов выберите соответствующий пункт контекстного меню класса. Все соединенные с данным классом связи также будут удалены.

Пример диаграммы классов приведен на рисунке 3.34.

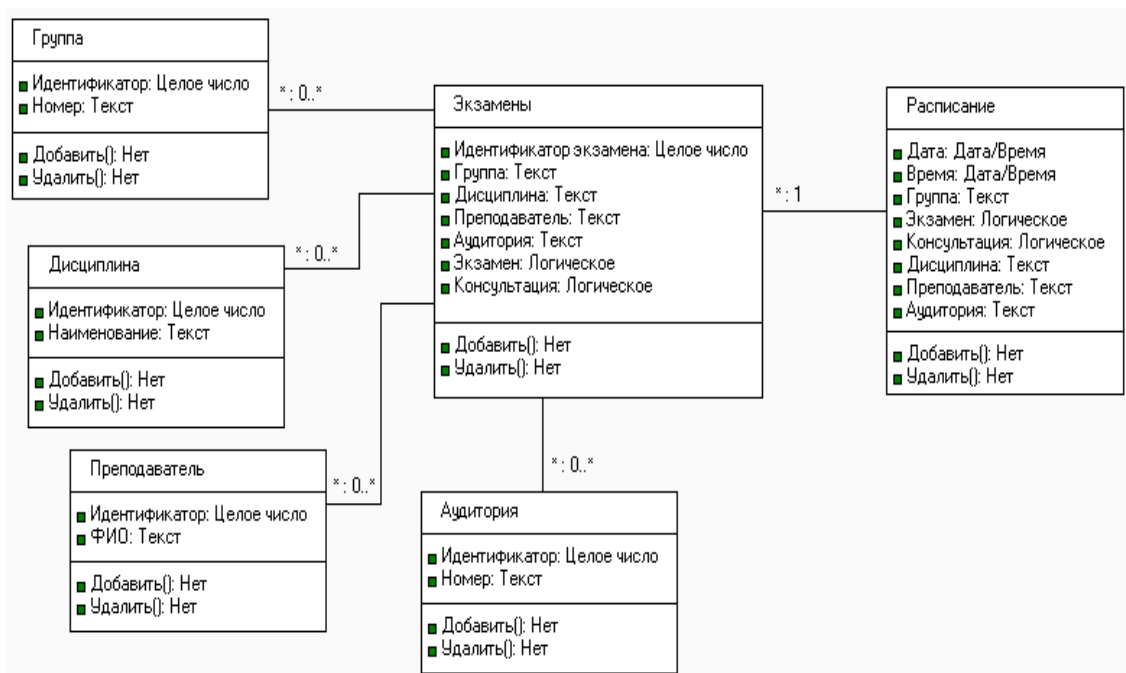


Рисунок 3.34. Диаграмма классов в «Bpsim.SD» для ИС «Расписание»

3.2.6. Подсистема моделирования пользовательского интерфейса

Моделирование пользовательского интерфейса возможно при наличии в проекте диаграммы последовательностей, и созданном на ней объекта-границы. В контекстном меню объекта-границы выбирается пункт «Форма». У каждой границы может быть только одна форма, поэтому по нажатию на соответствующий пункт меню откроется или новая форма, или уже созданная.

Менеджер пользовательских форм вызывается из контекстного меню объекта – границы диаграммы последовательности (рисунок 3.35).

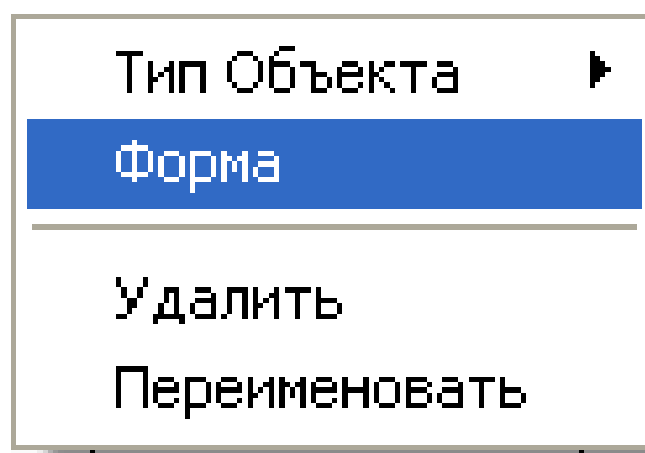


Рисунок 3.35. Контекстное меню объекта – границы

Графический интерфейс менеджера форм приведен на рисунке 3.36.

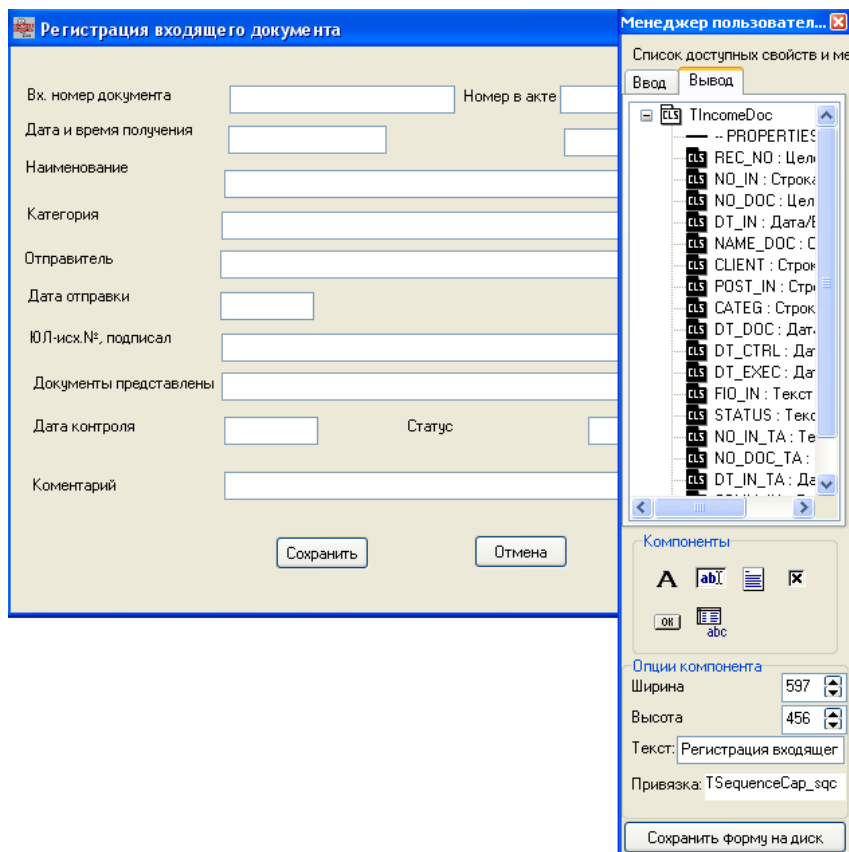








Рисунок 3.36. Форма «Менеджер пользовательских форм» и прототип формы

Менеджер форм содержит следующие элементы:

- Дерево входных и выходных классов проекта со списками доступных свойств и методов этих классов;
- Панель визуальных компонентов, размещаемых пользователем на проектируемой форме. Размещаемые компоненты могут иметь один из шести типов:
 - a)  Property Label – отображает не редактируемые однострочные текстовые свойства входных и выходных классов,
 - b)  Edit – отображает редактируемые однострочные текстовые свойства входных и выходных классов,
 - c)  Мемо – отображает многострочные текстовые свойства входных и выходных классов,
 - d)  Check Box – отображает булевы свойства входных и выходных классов,

е)  Button – кнопки, к которым осуществляется привязка методов входных и выходных классов,

ф)  String Grid – таблицы, полями которых могут быть свойства выходных классов;

– Панель свойств (опций) компонентов, помещенных на форму, содержит следующие элементы:

а) ширина и высота компонента. Допускается устанавливать данные параметры вручную или регулировать их с клавиатуры стрелками при нажатой клавише Shift,

б) текст, отображаемый в компоненте или заголовок,

с) привязка к конкретному свойству или методу выбранного класса. Для компонентов String Grid данный параметр отсутствует, его заменяет кнопка вызова «Редактора таблиц» (рисунок 3.37).

Если ни один компонент не выбран, на панели опций выводятся свойства проектируемой формы.

– Нажатие на кнопку «Сохранить форму на диск» формирует для спроектированной формы файлы *.pas и *.dfm и сохраняет их на компьютере в указанном месте. В дальнейшем эти файлы могут быть открыты в Delphi для дальнейшей доработки.

Редактор таблиц (рисунок 3.37) позволяет откорректировать порядок расположения и заголовки полей таблицы. Панель инструментов редактора содержит три кнопки: удаление поля, перемещение поля на одну позицию вверх и перемещение на одну позицию вниз.

Данные	Заголовок
Расписание.Дата	Дата
Расписание.Время	Время
Расписание.Группа	Группа
Расписание.Экзамен	Экзамен
Расписание.Консультация	Консультация
Расписание.Дисциплина	Дисциплина
Расписание.Преподаватель	Преподаватель
Расписание.Аудитория	Аудитория

Рисунок 3.37. Форма «Редактор таблиц»

Для размещения на форме элемента необходимо щелкнуть на выбранном компоненте на соответствующей панели «**Менеджера форм**» и на форме в нужном Вам месте. Выбранный в данный момент компонент выделяется синим цветом, а его свойства отображаются на панели опций (рисунок 3.38).

Рисунок 3.38. Панель опций менеджера форм

Перемещение компонентов осуществляется мышью с нажатой левой клавишей или стрелками с клавиатуры с нажатой клавишей Ctrl. Размеры компонентов можно изменять мышью, через панель опций или стрелками с клавиатуры с нажатой клавишей Shift. Удаление компонентов осуществляется через соответствующий пункт контекстного меню.

Чтобы привязать к компоненту определенное свойство или метод, необходимо «перетащить» это свойство (метод) из дерева классов на соответствующий компонент. При этом если данная привязка разрешена, то компонент выделяется красным цветом (рисунок 3.39).

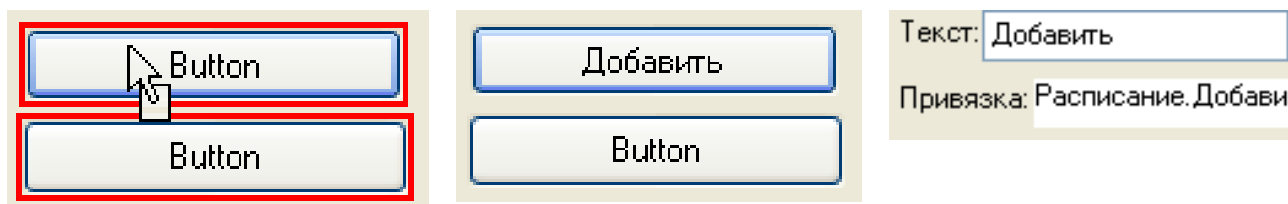


Рисунок 3.39. Привязка к кнопке метода «Добавить» класса «Расписание»

Если необходимо привязать к компоненту (таблице) все свойства определенного класса, нет необходимости привязывать каждое свойство в отдельности, достаточно «перетащить» имя класса на компонент String Grid.

Пример спроектированной формы «Расписание экзаменов» приведен на рисунке 3.40.

Рисунок 3.40. Спроектированная в «Bpsim.SD»
форма составления расписания экзаменов

3.3. Описание агента интеграции Bpsim.MAS и Bpsim.SD

Агент интеграции Bpsim.MAS и Bpsim.SD доступен пользователям системы BPSim.MAS (меню Общие/Агенты). На начальном этапе появляется форма, позволяющая выбрать необходимую модель предметной области (рис. 3.41).

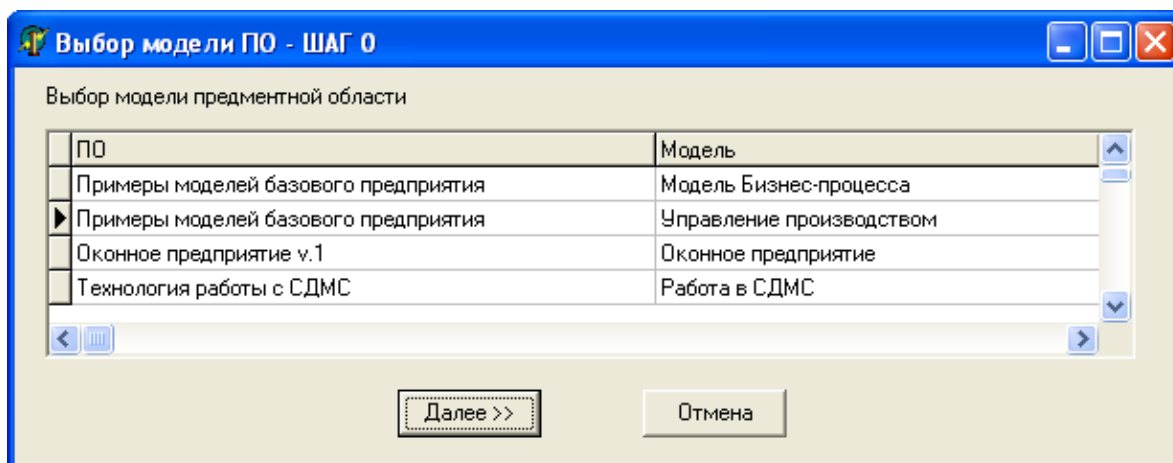


Рисунок 3.41. Форма первого этапа конвертации моделей

После этого предлагается отобразить операции модели (рис. 3.42).

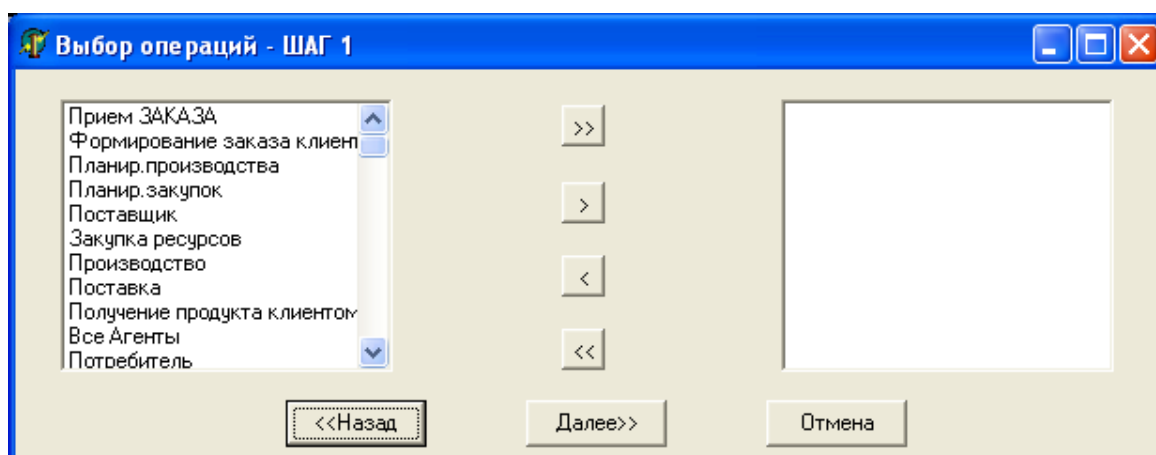


Рисунок 3.42. Форма второго этапа конвертации моделей

На следующем этапе агент последовательно конвертирует элементы модели МППР в соответствующие элементы модели ИС (см. таблицу 2.11).

3.4. Методика использования пакета Bpsim

В процессе бизнес-моделирования и проектирования ПО предметной области МППР используются следующие пакеты программ (www.bpsim.ru) [55]:

- Bpsim.MAS – мультиагентная система динамического моделирования ситуаций;
- Bpsim.MSS – интеллектуальная система технико-экономического проектирования;
- Bpsim.SD – CASE-средство.

Методика бизнес-моделирования и проектирования ПО предметной области МППР в стандарте IDEF0 показана на рисунке 3.43 [57; 58].

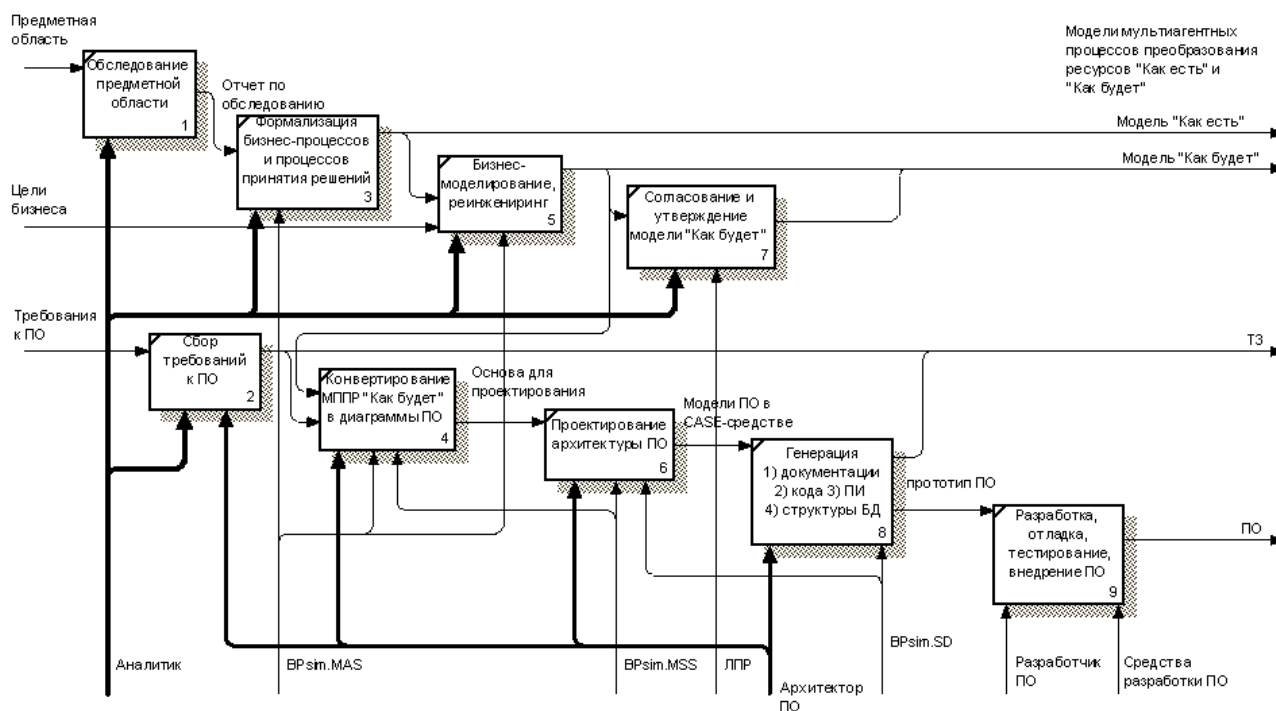


Рисунок 3.43. Методика бизнес-моделирования и проектирования ПО

По результатам обследования БП аналитик с помощью Bpsim.MAS создает модель БП предприятия «как было», проводит моделирование и, если это необходимо, строит модель «как будет». Результаты моделирования позволяют обосновать предложенные изменения в модели бизнеса. Результирующая модель «как будет» конвертируется в DFD-диаграммы пакета Bpsim.SD, учитываются только те процессы, которые будут автоматизированы. Этот пакет используется для проектирования ИС. На основе данных DFD-диаграмм могут быть получены UML-диаграммы прецедентов, а также шаблоны классов и диаграмм последовательности, заготовка структуры БД. Затем архитектор ПО строит диаграммы последовательностей и классов, а разработчик ПИ – шаблоны форм. Данные заготовки будущей ИС могут быть преобразованы в программный код, который дорабатывается программистами.

Программная реализация интеграции Bpsim.MAS и Bpsim.SD возможна благодаря тому, что данные этих двух продуктов хранятся в единой БД. Конвертацию информации из одной системы в другую осуществляют диалоговые программные агенты-помощники.

Агенты-помощники, входящие в состав пакета Bpsim, выполняют следующие функции:

- осуществляет передачу информации между программными продуктами в рамках комплексного решения единой задачи;
- облегчает работу непрограммирующего пользователя в процессе ознакомления с продуктами семейства Bpsim;
- реализует функцию валидации на этапах создания модели МППР, проектирования КМПО ИС.

DFD-диаграмма, отражающая последовательность действий агента-помощника при конвертации модели МППР в модель ИС, представлена на рисунке 3.44.

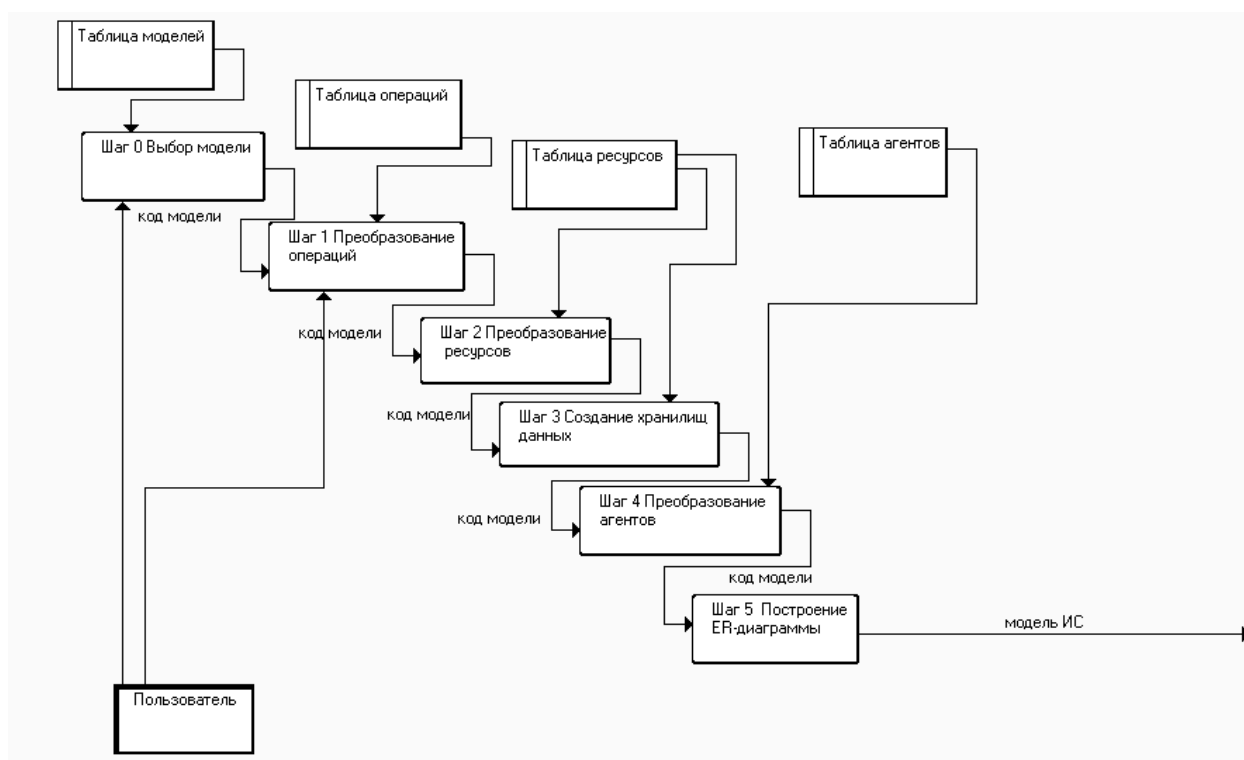


Рисунок 3.44. DFD-диаграмма преобразования модели МППР в модель ИС

Преимущество метода интеграции программных продуктов при помощи агентов-помощников заключается в возможности для непрограммирующего пользователя комплексно решать задачи моделирования бизнеса и разработки ИС. Кроме того, интеграция дает возможность существенно упростить и ускорить работу аналитиков и проектировщиков.

Данная методика и продукты линейки Bpsim позволяют комплексно решать задачи моделирования бизнеса, технико-экономического проектирования, моделирования архитектуры ИС и разработка прототипа приложения, что в итоге позволяет существенно упростить и ускорить работу аналитиков, а также повысить качество готового ПО за счет автоматизации некоторых этапов и сокращения влияния человеческого фактора.

Выводы по главе 3

Решение задачи интеграции имитационного, экспертного, ситуационного и мультиагентного моделирования, а также функционального и объектно-ориентированного подхода, позволило реализовать СППР в области разработки ИС Bpsim.SD.

В третьей главе дано подробное описание возможностей Bpsim.SD: функциональная и ОО разработка ИС, моделирование ПИ. Приводится методика использования пакета Bpsim и описания агента-помощника для преобразования модели МППР в модель ИС.

4. Применение СППР Bpsim.SD при разработке информационных систем

4.1. Проект по анализу бизнес-процессов и разработке технического задания на единую информационную систему вуза УГТУ-УПИ

До 2010 года УрФУ назывался Уральский государственный технический университет – УПИ имени первого Президента России Б. Н. Ельцина. ВУЗ представляет собой предприятие со сложной организационной структурой (факультеты, кафедры, филиалы и представительства), поэтому для успешной автоматизации бизнес-процессов ВУЗа УГТУ-УПИ необходимо было провести полное обследование того, как они протекают. В начале проекта были проведены интервью с сотрудниками и представителями руководства ВУЗа. Были проведены интервью трех проректоров, сотрудников и руководителей шести деканатов (радиотехнический институт, физико-технический факультет, химико-технологический факультет, механико-машиностроительный факультет, Институт образовательных технологий, заочного факультета), начальника отдела информационных технологий Института информатики радиоэлектроники и связи, начальника Административного отдела, начальника отдела инновационно-образовательных технологий, заместитель начальника Административного отдела по архивной работе, ученых секретарей кафедр, директора Регионального центра новых информационных технологий, заведующего отделом информационных технологий зональной научной библиотеки, начальника и сотрудников отдела автоматизированных систем управления (АСУ), сотрудников личного стола студентов (ЛСС) [86].

Анализ показал, что учебный процесс частично автоматизирован, но существующие ИС не всегда связаны друг с другом, содержат дублирующую информацию. Учебный процесс связан с подготовкой большого количества первичной документации и отчетов, в подготовке одного типа данных (стипендия, приказы и т. п.) участвует большое количество структурных подразделений ВУЗа [87].

Результаты обследования учебного процесса с использованием структурного и объектно-ориентированного анализа выполнялись в рамках НИР № *01200601073 [87, 88]. В результате анализа БП с помощью CASE-средства Bpsim.SD [86,89-93] было построено 60 IDEF0 диаграмм.

Обследование БП выявило неоптимальность выполнения некоторых процессов УГТУ-УПИ, в частности «Ход сессии» и «Движение контингента». Рассмотрим подробнее каждый из них [87].

4.1.1. Процесс «Ход сессии»

Процесс «Ход сессии» включает в себя формирование экзаменационных и зачетных ведомостей, выдачу и регистрацию зачетных и экзаменационных листов, прием экзаменов и зачетов в сессию и пересдач, ввод результатов зачетов в зачетную неделю, экзаменов в сессию и пересдач в сессию и вне сессии, формирование отчетов по аттестации студентов. В процессе «Ход сессии» задействованы сотрудники деканата, отдел автоматизированных систем управления, учебно-методическое управление (УМУ), планово-финансовое управление (ПФУ). Основная нагрузка ложится на сотрудников деканата, которые вводят исходную информацию и формируют отчеты по итогам сессии. Работу, связанную с формированием экзаменационных и зачетных ведомостей и контролем итогов сессии, выполняет отдел АСУ. Повторная сдача зачетов и экзаменов осуществляется по экзаменационным и зачетным листам, которые выдают заместители деканов. Отделы УМУ и ПФУ предоставляют первоначальную информацию – учебные планы, стипендиальный фонд для распределения стипендии. Процесс «Ход сессии» сопровождается подготовкой большого количества бумажных документов, что в свою очередь сильно замедляет распространение информации до заинтересованных лиц. Выявленные проблемы позволили сформулировать требования к автоматизации процесса «Ход сессии» [94].

Было предложено внести изменения в процесс работы сотрудников деканата (рисунок 4.1): бумажное заполнение семестровых журналов

заменяется вводом данных в электронную версию семестрового журнала (аналогично протоколы стипендий), а бумажный носитель получать в процессе печати. Появление единой БД снимает проблему оперативной передачи информации (отрывные талоны, коррекция оценок по итогам сессии, заполненные протоколы на стипендию, итоги зачетной недели в УМУ, сводные таблицы для коррекции оценок по итогам сессии, протоколы на стипендию от отдела АСУ, промежуточные и итоговые результаты сессии от УМУ). Выдача экзаменационных листов будет регистрироваться в электронном журнале, а отметки возврата и выдачи в семестровом журнале.

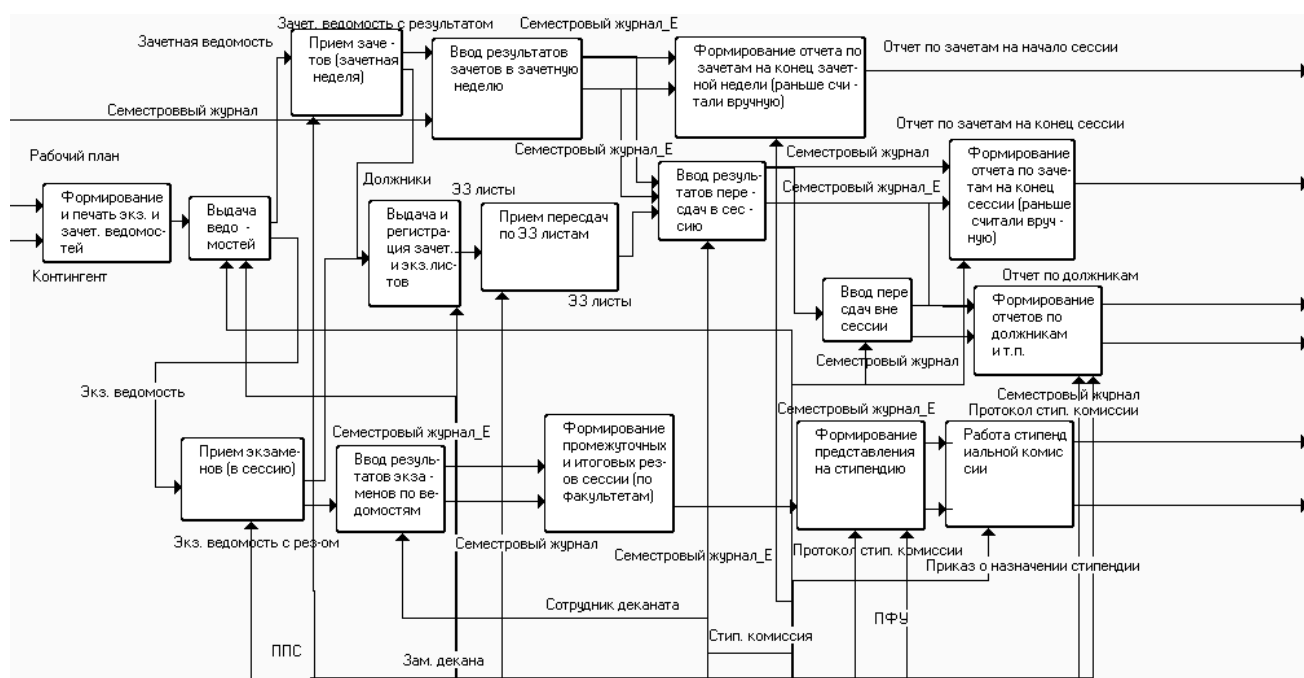


Рисунок 4.1. Модель «как будет» процесса «Ход сессии»

Таким образом, автоматизация процесса «Ход сессии» (модель «как будет») позволит: сформировать актуальную БД, автоматизировать выдачу и учет экзаменационных и зачетных листов, вводить результаты сдач и пересдач в электронный журнал, автоматизировать формирование приложений к диплому, актуальной академической справки, отчета по задолженностям (должникам) и отчетов по зачетам, параллельно работать с семестровым журналом, автоматически формировать актуальное приложение к диплому.

4.1.2. Процесс «Движение контингента»

Процесс «Движение контингента» включает в себя формирование выписок в приказ, списочных приказов, сборных приказов по УГТУ – УПИ, формирование различных справок («Справка студенту», «Справка о том, что является студентом», «Справка о стипендии» и т. д.). Основными участниками этого процесса являются сотрудники деканатов и ЛСС, деканы факультетов, проректор. Процесс был организован таким образом, что с момента получения сотрудником деканата первичных документов до момента изменения информации о студенте в БД и получении выписки из приказа проходило много времени. Кроме того, участники процесса выполняли много рутинной работы:

- сотрудники деканата формировали в разное время два схожих по содержанию документа – заявление (помогали написать студенту) и выписку в приказ;
- проректор три раза просматривал и визировал различные документы о студенте (заявление, выписку в приказ, приказ).

В результате анализа были предложены следующие изменения (рисунки 4.2-4.5):

- во время написания заявления студентом сотрудник деканата создает его электронный аналог в единой БД;
- предлагается отказаться от формирования выписки в приказ, вся необходимая информация для издания приказа заносится в единую БД, следовательно, сбор виз будет проводиться один раз;
- сотрудники ЛСС будут работать с электронной формой заявления и смогут сами его корректировать, т. е. не будет теряться время на отправку документов в деканат для переоформления;
- появится возможность формировать и хранить дела студентов в электронном виде (история по студенту);
- у проректора будет подписываться только заявление (для некоторых видов заявлений достаточно визы декана) и приказ по УГТУ-УПИ;

- в информационной системе можно будет отслеживать маршрут и стадии прохождения документов по процессу;
- у сотрудников отдела АСУ исчезнет не свойственная им функция – выполнение проводок приказа в базе данных;
- в случае списочного приказа, соответствующие информационные объекты в информационной системе формирует сотрудник деканата, в дальнейшем с ними работает сотрудник ЛСС.

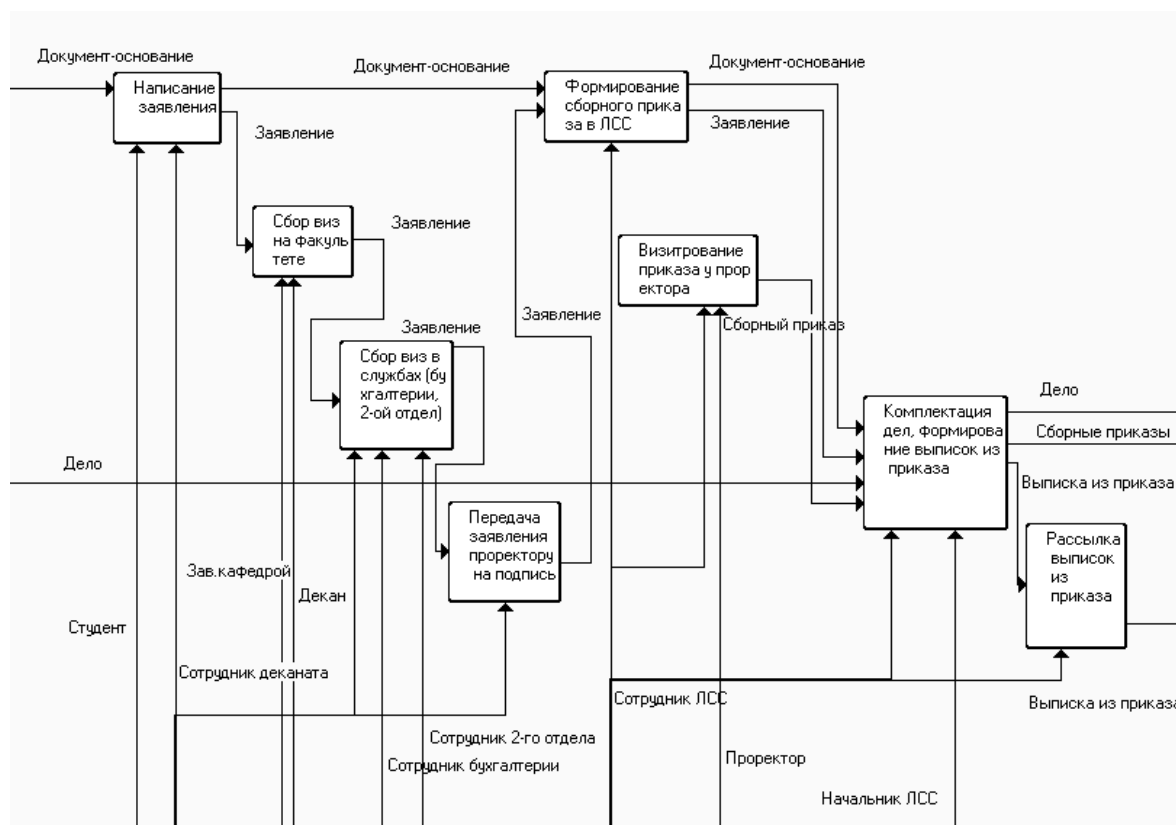


Рисунок 4.2. Модель «как будет» процесса «Движение контингента»

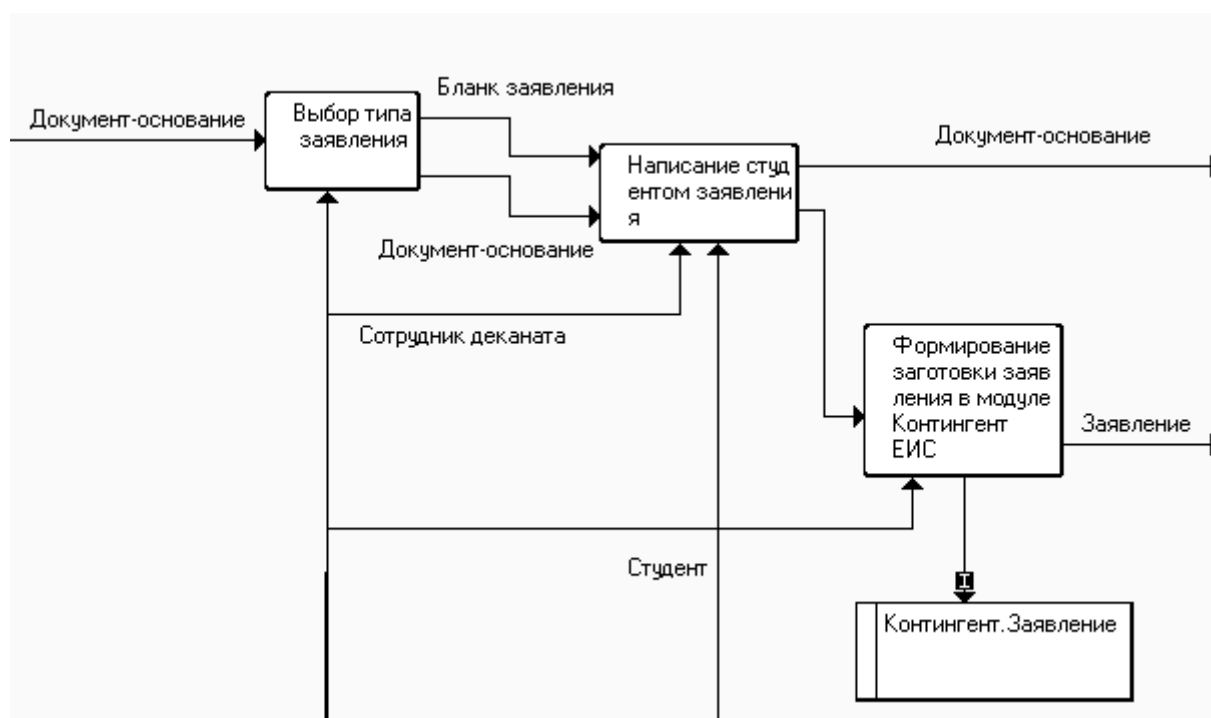


Рисунок 4.3. Написание заявления

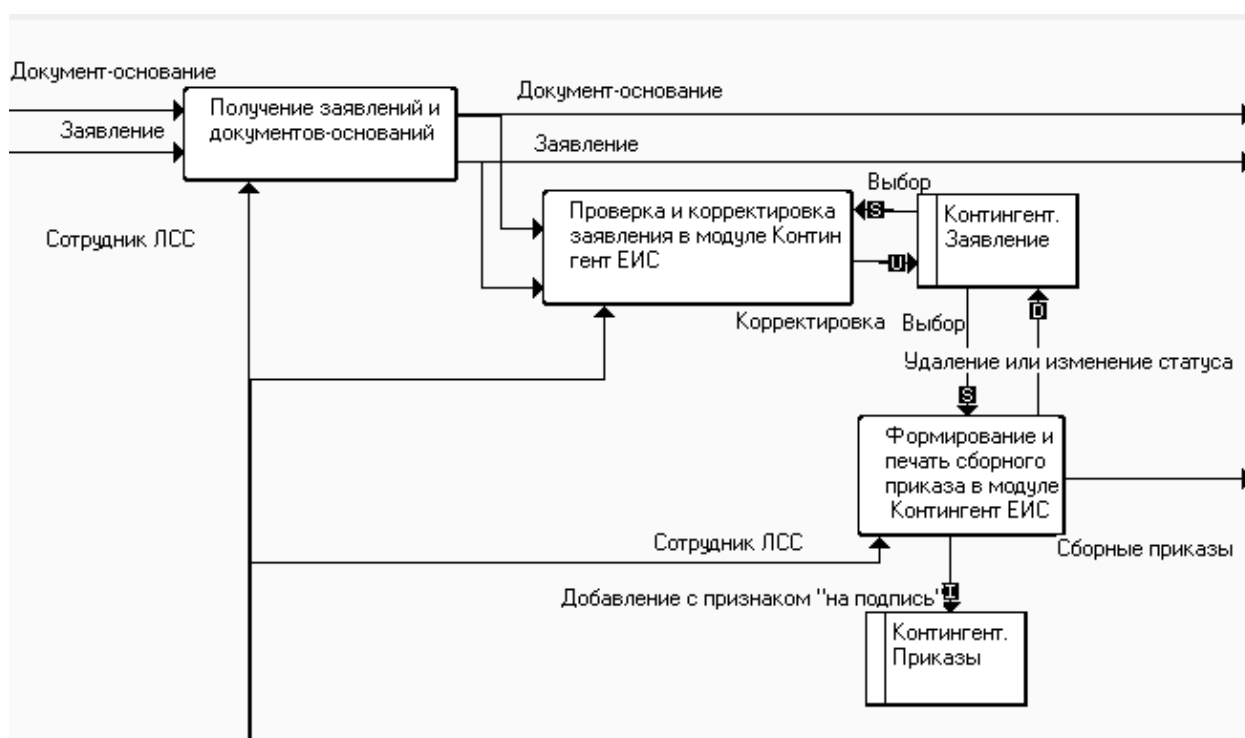


Рисунок 4.4. Формирование сборного приказа в ЛСС

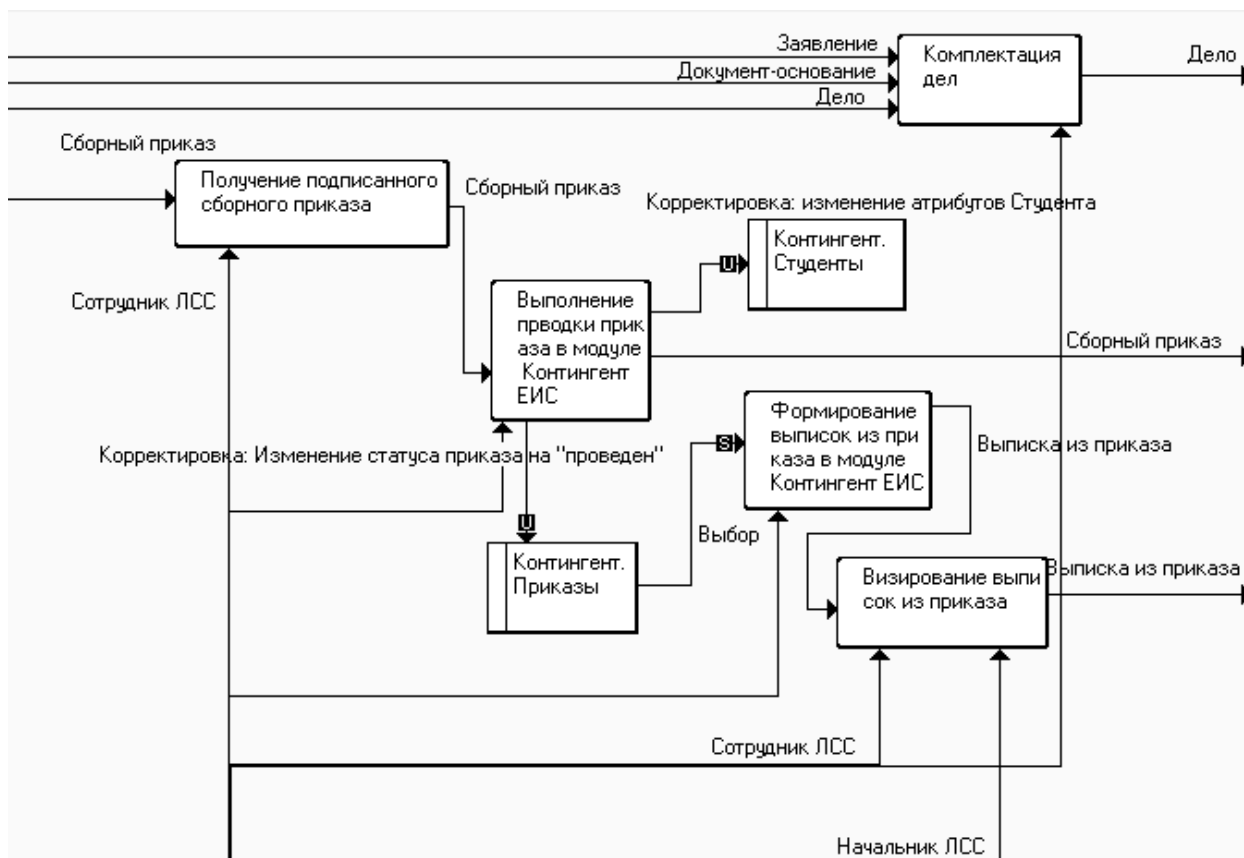


Рисунок 4.5. Комплектация дел, формирование выписок из приказа

Таким образом, автоматизация процесса «Движение контингента» (модель «как будет») позволит сократить временные трудовые затраты сотрудников ВУЗа и оперативно получать достоверную информацию о студентах.

По результатам обследования было написано ТЗ на разработку ЕИС ВУЗа [95]. Оно включает в себя требования и диаграммы, описывающие архитектуру системы. По соглашению с Заказчиком использовались DFD-диаграммы, UML-диаграммы прецедентов, последовательности и классов. ТЗ содержит 25 DFD-диаграмм, 14 диаграмм прецедентов, 18 диаграмм последовательности и 1 диаграмму классов. Например, функциональность модуля Расписание ЕИС ВУЗа показана на рисунке 4.6.

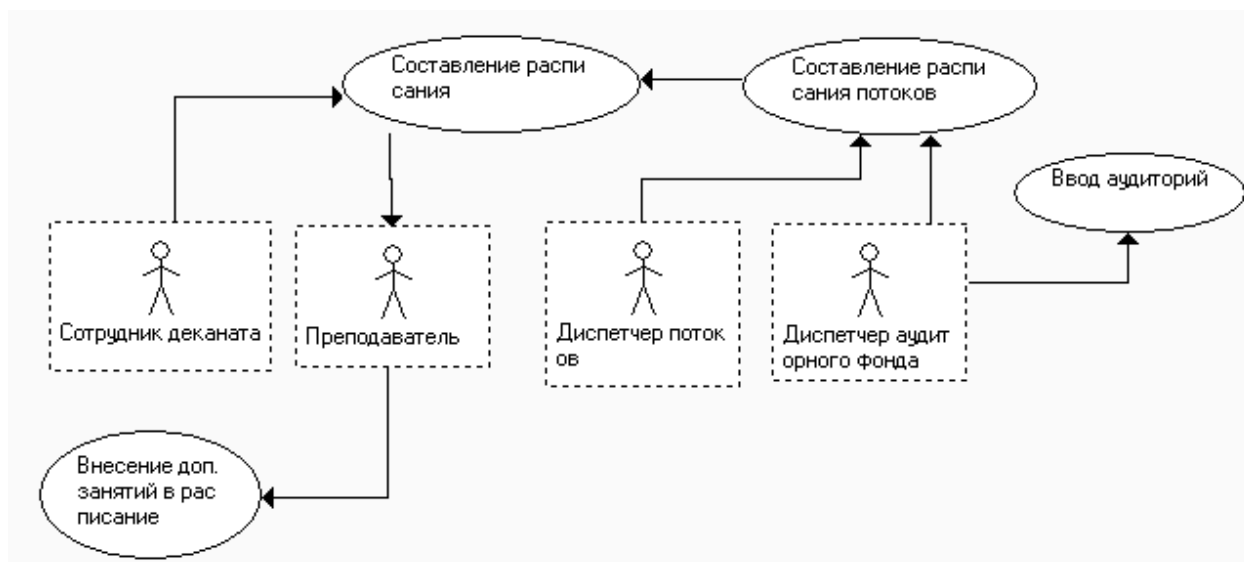


Рисунок 4.6. Диаграмма вариантов использования модуля Расписание

Последовательность действий для прецедента «Составление расписания учебных занятий/экзаменов» показана с помощью диаграммы последовательности на рисунке 4.7.

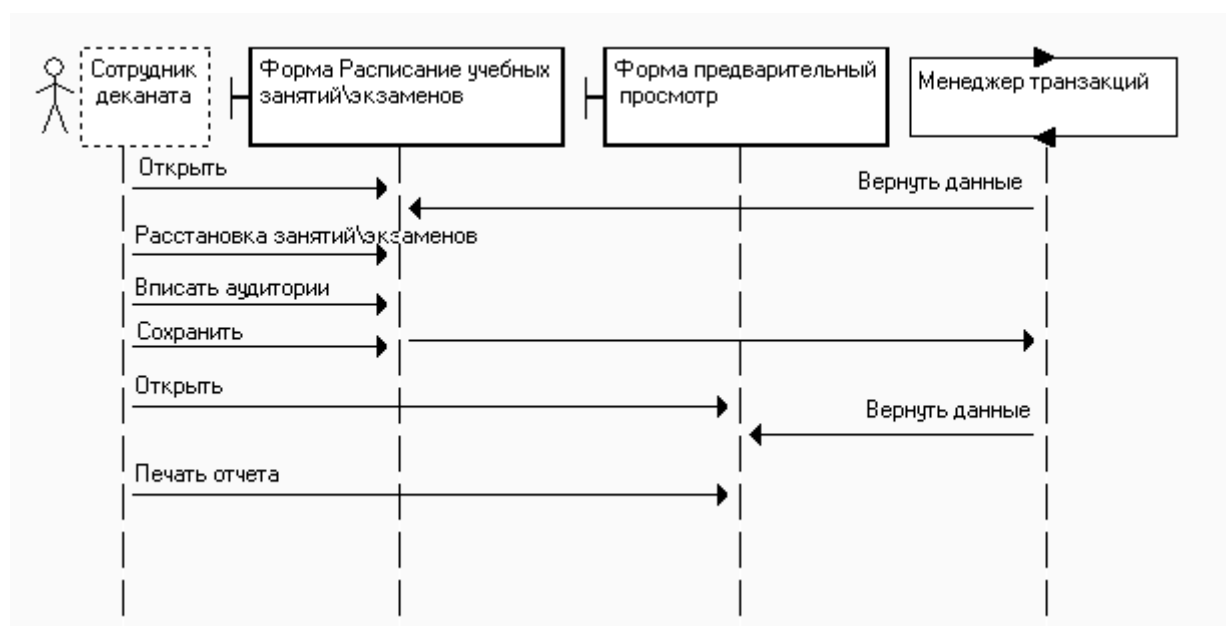


Рисунок 4.7. Диаграмма последовательности для прецедента «Составление расписания учебных занятий/экзаменов»

Bpsim.SD позволяет на основании данных о бизнес-объектах ИС осуществлять проектирование прототипов экранных форм проектируемой системы [86; 96], которые также были включены в ТЗ [95]. На рисунке 4.8. приведен пример спроектированной формы ввода расписания.

Рисунок 4.8. Спроектированная форма ввода расписания экзаменов в Bpsim.SD

4.1.3. Оценка эффективности внедрения модуля «Движение контингента»

Для оценки эффективности внедрения модуля «Движение контингента» ЕИС университет была построена имитационная модель процесса движения контингента «как было» и «как будет» в системе динамического моделирования ситуаций (СДМС) Bpsim.MAS. Данные для модели были получены в результате опроса сотрудников ЛСС, отдела АСУ и четырёх деканатов, которые полгода проработали с новым модулем ЕИС.

На рисунке 4.9 показана модель процесса движения контингента «как было» и «как будет». С помощью переключателя, расположенного справа (рисунок 4.9), можно выбрать вариант проигрываемой модели: модели «как было» или модели «как будет», который определяет время работы каждого узла. Узлы модели представляют собой этапы обработки документов. Узлы 1-10 моделируют обработку документа на факультетах, узлы 11-14 – обработку в ЛСС накопившихся за неделю документов, подготовленных факультетами. В результате автоматизации процесса движения контингента исчезла необходимость прохождения документом некоторых этапов обработки, поэтому при выборе модели «как будет» в имитации не участвуют узлы с номерами 7, 9, 10, 15 (рисунок 4.9).

Движение контингента

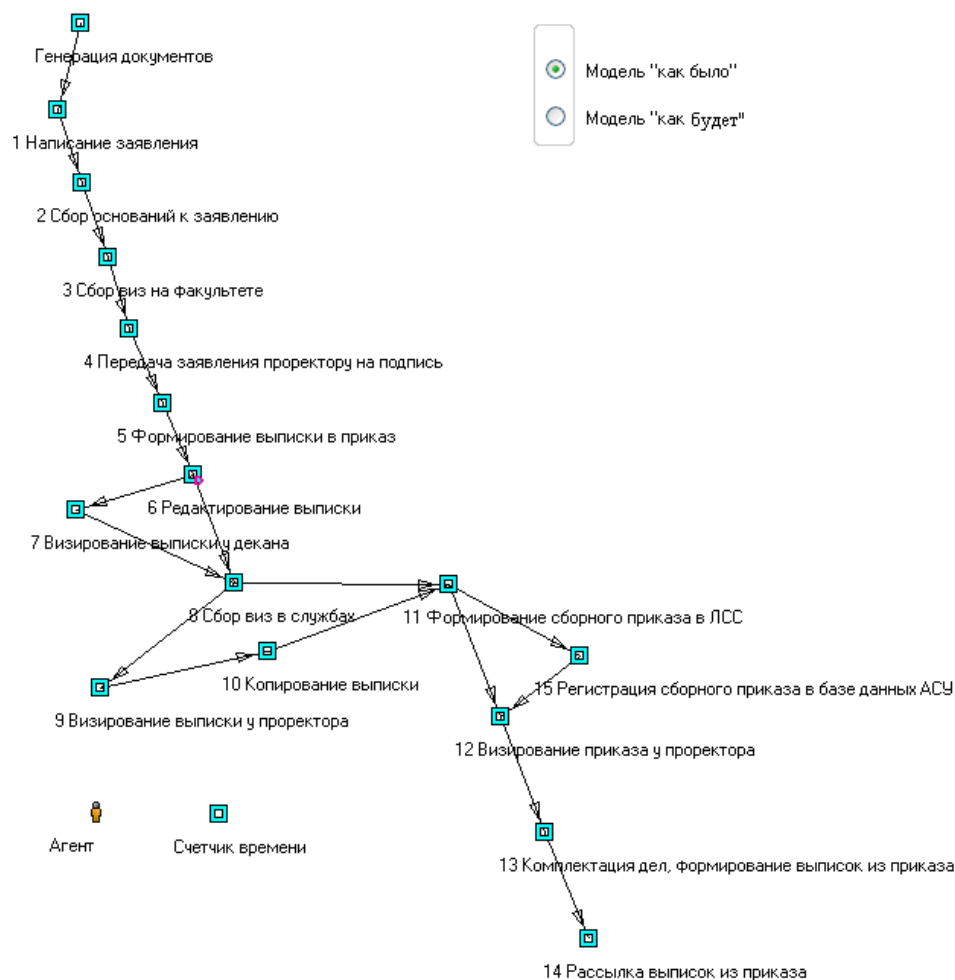


Рисунок 4.9. Модель процесса движение контингента

Затем была промоделирована работа деканатов и ЛСС в течение месяца (160 ч) в двух вариантах: до и после внедрения модуля «Движение контингента». Результаты проведения экспериментов представлены в виде графиков.

График «Количество обработанных документов» (рисунок 4.10) показывает максимальное при заданных условиях количество документов, прошедших все этапы от написания заявления студентом, до рассылки выписки из приказа сотрудниками ЛСС.

График «Накопление документов для пачки» (рисунок 4.11) отображает процесс накопления за неделю на факультетах документов, из которых в ЛСС формируется пачка.

График «Накопление документов для пачки» (рисунок 4.12) отображает процесс накопления за неделю на факультетах документов, из которых в ЛСС формируется пачка (сборный приказ).

Внедрение модуля «Движение контингента» позволило избежать потери документов в ЛСС, в отделе АСУ.

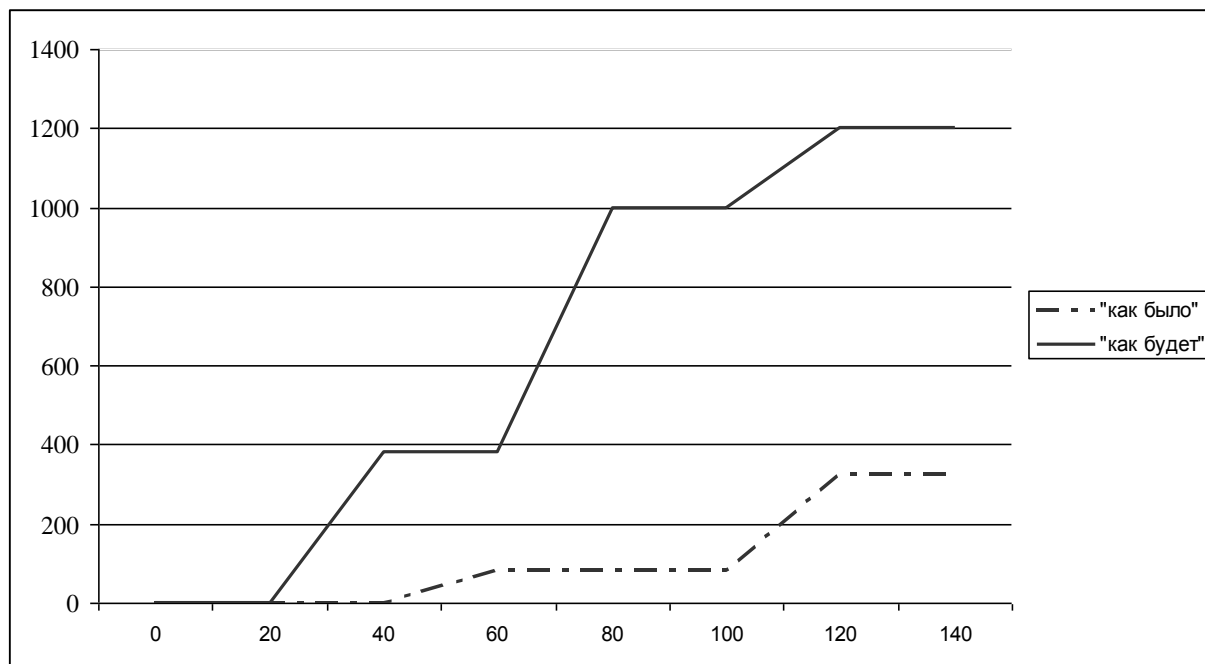


Рисунок 4.10. Количество обработанных документов за период

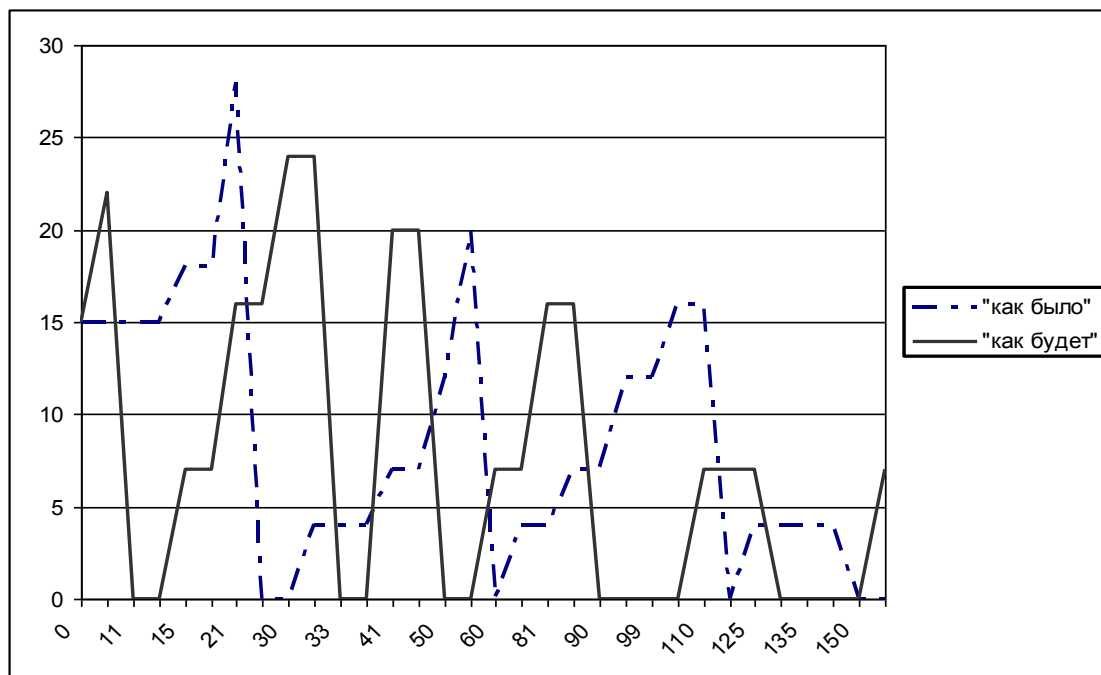


Рисунок 4.11. Накопление документов для пачки

Обобщенные результаты экспериментов приведены в таблице 4.1.

Таблица 4.1

Оценки эффективности внедрения модуля «Движение контингента»

ЕИС университет

Показатели	Модель «как было»	Модель «как будет»
Количество обрабатываемых документов за месяц, шт.	390	1264
Количество потерянных документов за месяц, шт.	12	0
Производительность труда сотрудников деканата, документ/ч	0,4	0,5
Производительность труда сотрудников ЛСС, документ/ч	2,4	7,9

Таким образом, благодаря автоматизации процесса «Движение контингента» производительность сотрудников деканата повысилась на 25%, а производительность сотрудников ЛСС повысилась на 229% (т. е. более чем в 3 раза). Экономический эффект от внедрения предложенных моделей «как будет» и автоматизации процесса «Движение контингента» составляет 1 027 тыс. руб. в год.

4.2. Разработка дополнительных модулей для системы ведения реестров акционеров «Вереком-2»

СВР «Вереком-2» предназначена для автоматизации деятельности профессионального участника рынка ценных бумаг – регистратора. Для анализа деятельности регистратора была построена имитационная модель в СДМС Vpsim.MAS. Были проведены имитационные эксперименты для определения оптимального количества операторов, занимающихся регистрацией входящих документов для ЗАО «Ведение реестров компаний». Для проверки работы предложенного метода была проделана работа по созданию части ИС, отвечающей за регистрацию входящих документов [85].

4.2.1. Регистрация входящих документов

Модель процесса регистрации входящих документов показана на рисунке 4.12. Далее она была конвертирована в Vpsim.SD, в результате процесс

регистрации входящих документов был представлен DFD-диаграммой (рисунок 4.13).

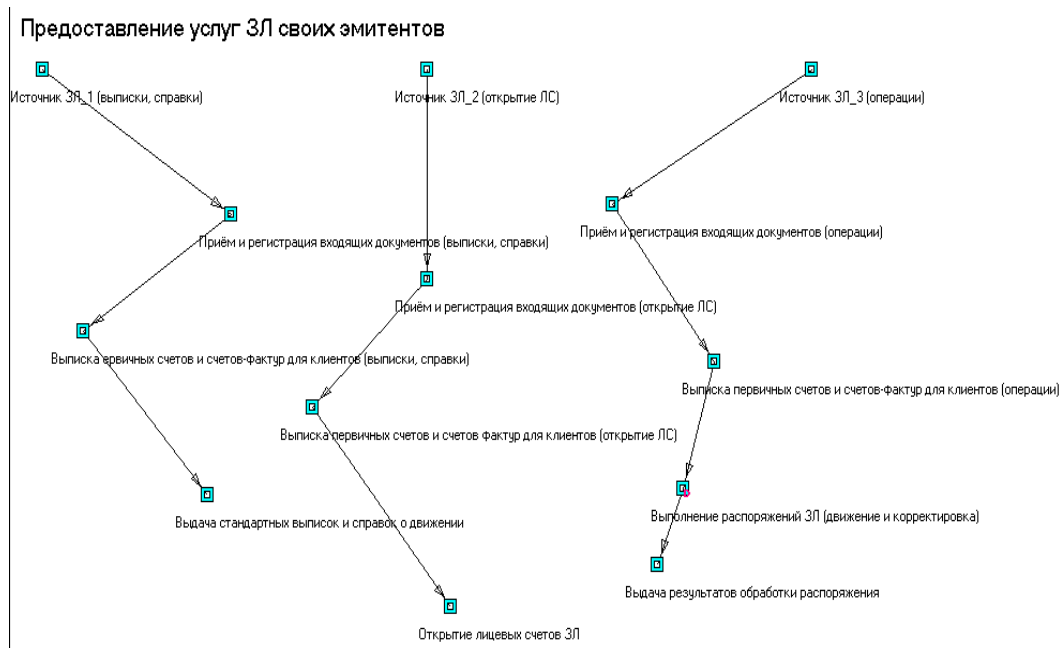


Рисунок 4.12. Модель процесса регистрации входящих документов

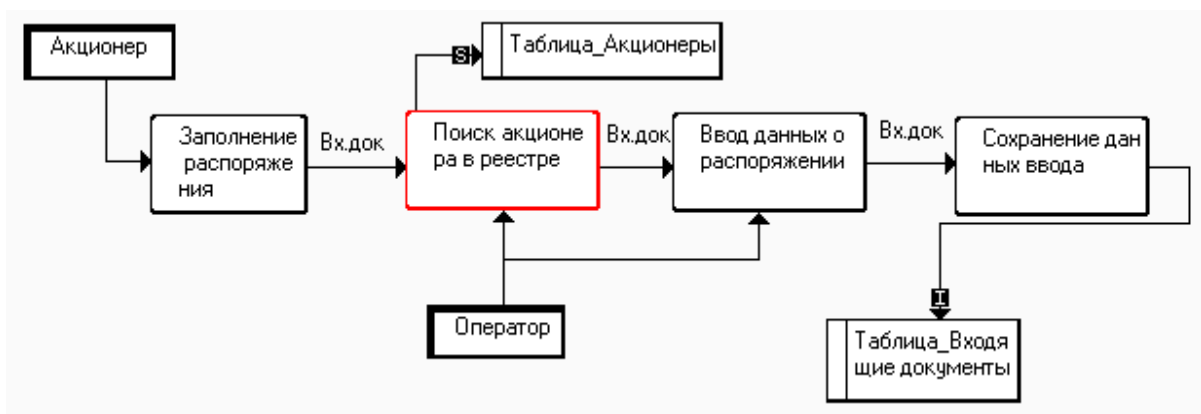


Рисунок 4.13. DFD-диаграмма процесса регистрации входящего документа

Также были получены диаграмма вариантов использования подсистемы «Регистрация входящих документов» (рисунок 4.14) и диаграмма основных классов (рисунок 4.15).

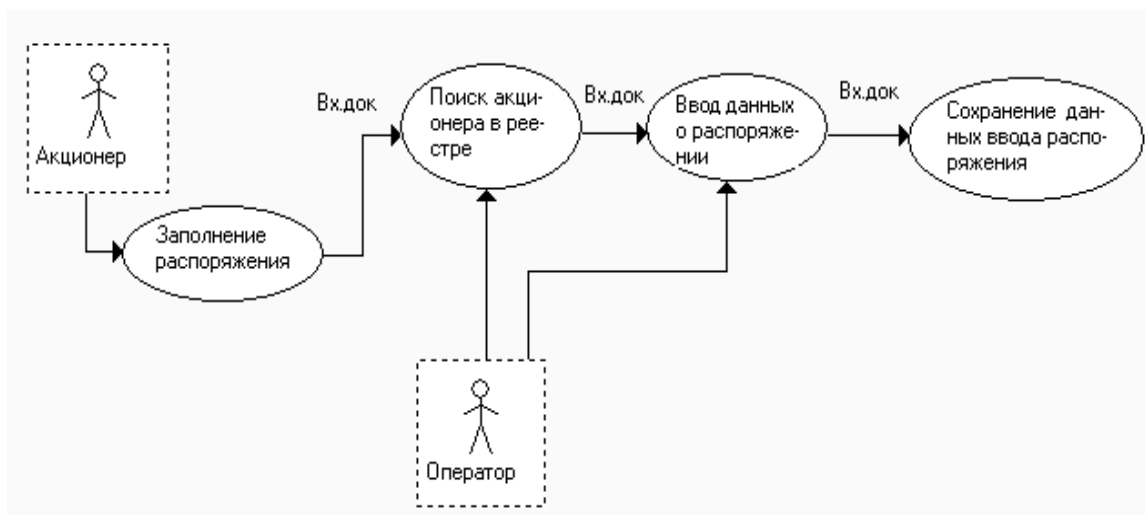


Рисунок 4.14. UML-диаграмма прецедентов процесса регистрации входящего документа

Далее были построены диаграмма последовательности регистрации входящего документа (рисунок 4.16) и диалоговая форма (рисунок 4.17).

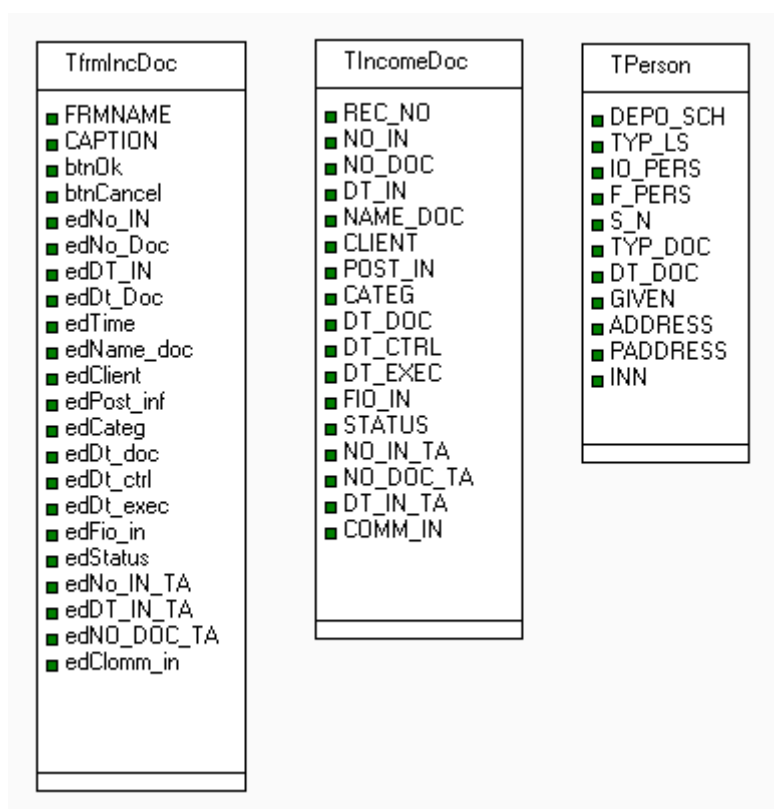


Рисунок 4.15. Основные классы



Рисунок 4.16. UML-диаграмма последовательности процесса
«Регистрация входящего документа»

Регистрация входящего документа

Вх. номер документа	<input type="text"/>	Номер в акте	<input type="text"/>
Дата и время получения	<input type="text"/>		<input type="text"/>
Наименование	<input type="text"/>		
Категория	<input type="text"/>		
Отправитель	<input type="text"/>		
Дата отправки	<input type="text"/>		
ЮЛ-исх. №, подписал	<input type="text"/>		
Документы представлены	<input type="text"/>		
Дата контроля	<input type="text"/>	Статус	<input type="text"/>
Комментарий	<input type="text"/>		

Рисунок 4.17. Шаблон диалоговой формы «Регистрация входящего документа»

Диалоговая форма реального приложения представлена на рисунке 4.18.

Добавление нового единичного документа

Вх.№ документа: 170/ТХК № в акте:

Дата и время получения: 23.06.2008 15:02:34

Наименование: Передаточное распоряжение

Категория: Документ (основание для провед. операции)

Отправитель:

Дата отправки: . . 15

ЮЛ-исх №, подписал:

ФЛ-почтовый адрес:

ФИО представителя:

Документы представлены: Уполномоченным представителем

Категория отправителя: Зарегистрированное лицо

Дата контроля: 26.06.2008 Срок: 3

Статус: Обрабатывается

Комментарий:

Вх № ТА:

Дата приема в ТА: . . 15 Дата отправки регистратору: . . 15

☐ Отображать только незаполненные акты

Создать единичный Создать активированный Копировать Сохранить Закрыть

Рисунок 4.18. Диалоговая форма регистрации вх. документа

В результате проектирования отдельной функции СВР «Вереком-2» в Bpsim.SD удалось описать весь БП регистрации входящих документов, описать основные классы и получить форму ПИ. Полученные заготовки программных модулей доработаны до функциональности, аналогичной той, которая уже реализована в СВР «Вереком-2», что и подтверждает работоспособность пакета и метода.

4.2.2. Модуль «Трансфер-агентский обмен»

Функционирование учетной системы фондового рынка сложно представить без института трансфер-агента (ТА). ТА пункты выполняют следующие функции:

- принимают от акционеров и передают регистратору информацию, необходимую для исполнения операций в реестре акционеров;
- принимают от регистратора и передают акционеру информацию из реестра (выписки, справки, уведомления).

Модуль ТА обмена предназначен для автоматизации этих функций. Этот модуль представляет собой программного агента, обладающего свойствами сотрудничества и коммуникативности. Два агента обмениваются между собой информацией по средствам передачи сообщений определенного формата. В заголовке сообщения указывается, от какого ТА пункта по какому эмитенту какая информация поступила. Основываясь на этих данных, принимаемая сторона выполняет определенные действия. Рассмотрим подробнее принцип взаимодействия двух агентов.

Перед первым взаимодействием им необходимо обмениваться служебными сообщениями, содержащими следующую информацию: открытые ключи шифрования; данными по эмитентам, по которым будет осуществляться обмен информацией; данные по выпускам ценных бумаг этих эмитентов. После обмена служебными сообщениями агенты готовы к взаимодействию.

Пользователь с помощью модуля Ведения реестров формирует информационные сообщения, которые сохраняются в очереди исходящих сообщений. ТА модуль периодически опрашивает состояние очереди. Обнаруженные сообщения шифруются, архивируются и посылаются по электронной почте. Полученные по электронной почте сообщения помещаются ТА модулем в очередь входящих сообщений. Затем они разархивируются, расшифровываются и распределяются по нужным реестрам. Отработанные сообщения поступают в архив.

Диаграмма прецедентов для программного агента ТА представлена на рисунке 4.19. Диаграммы последовательности приема и отправки почтовых сообщений представлены на рисунках 4.20 и 4.21.

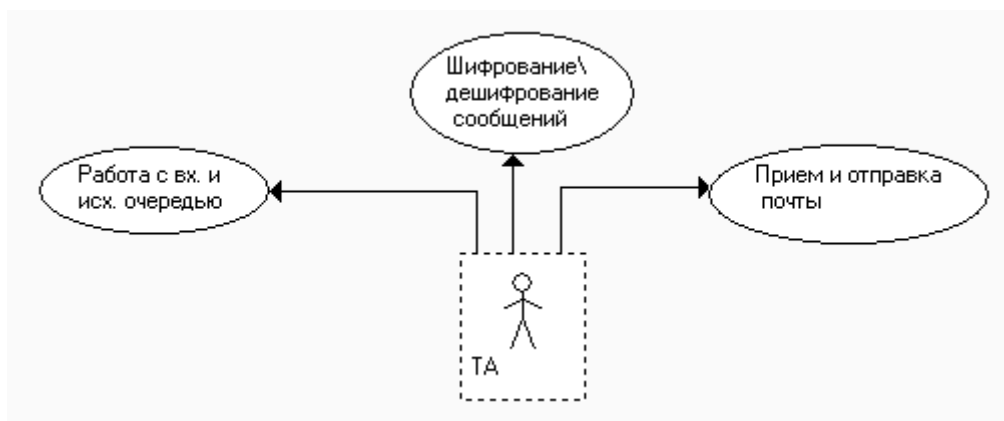


Рисунок 4.19. Диаграмма прецедентов программного агента ТА



Рисунок 4.20. Диаграмма последовательности отправки почтовых сообщений

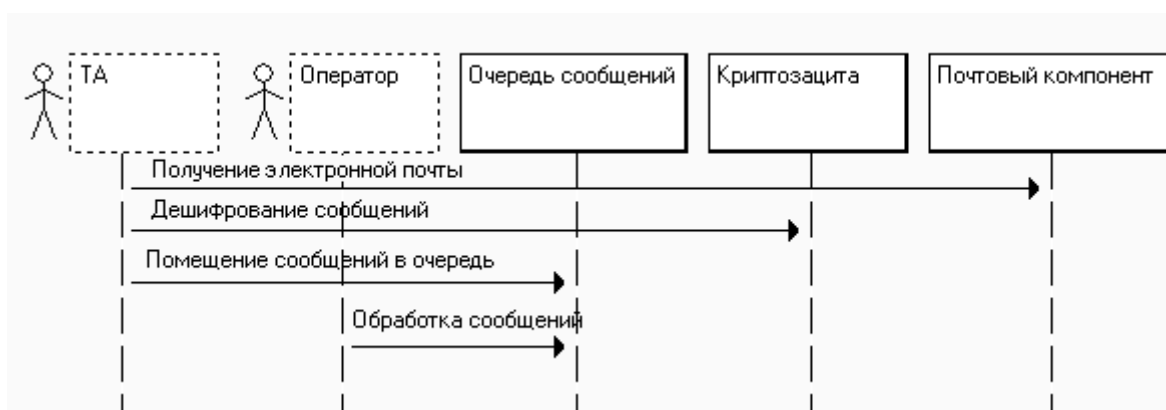


Рисунок 4.21. Диаграмма последовательности отправки почтовых сообщений

В результате проектирования программного агента в Bpsim.SD [85] удалось описать всю необходимую информацию для его реализации. Полученный ТА-модуль СВР «Вереком» выполняет функции прием и отправки почты, работу с очередями распоряжений, т. е. выполняет необходимые действия для организации ТА обмена.

4.2.3. Система «Электронный документооборот»

Для улучшения качества работы регистратора на фондовом рынке необходимо внедрение электронного документооборота (ЭДО) между регистратором и депозитарием. Профессиональная Ассоциация Регистраторов Депозитариев и Трансфер-Агентов (ПАРТАД) разработала форматы электронного взаимодействия регистраторов и депозитариев [97], которые рекомендовалось использовать при организации ЭДО.

Дополнительный модуль СВР «ВЕРЕКОН-2» – «Электронный документооборот» позволяет автоматически принимать электронные документы (ЭД) в базу данных СВР «ВЕРЕКОН-2» и формировать электронные выписки, справки и уведомления [85]. Для подтверждения приема ЭД формируется техническое сообщение (ТС), в котором указывается успешность приема либо возникшие при приеме ошибки. DFD-диаграмма процесса приема ЭД показана на рисунке 4.22.

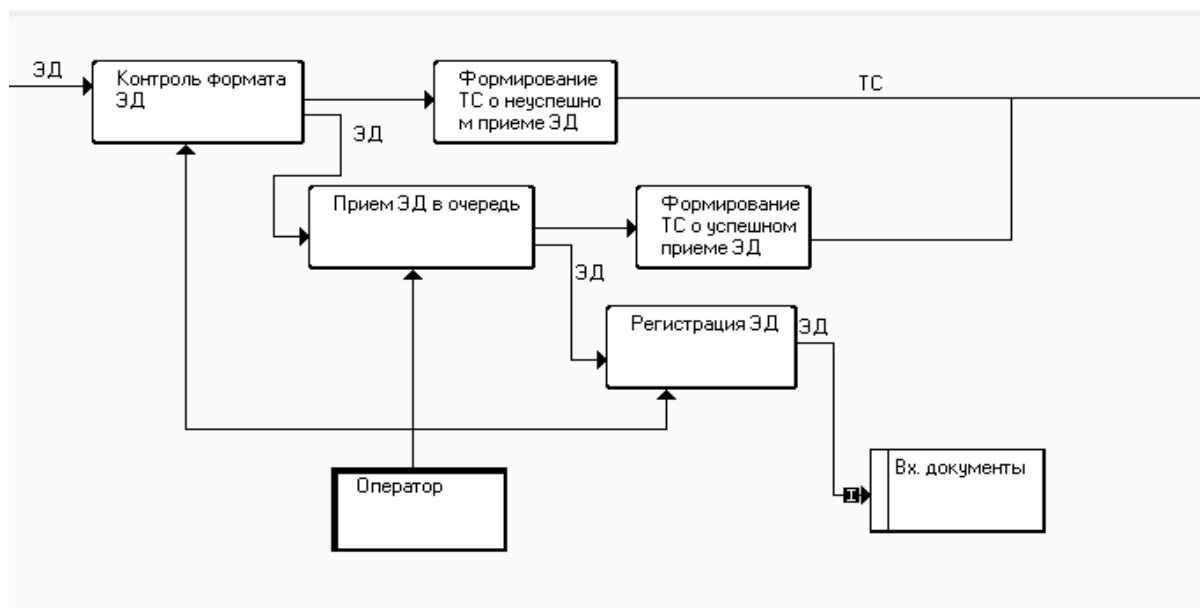


Рисунок 4.22. DFD-диаграмма процесса приема электронных документов

В рассматриваемом процессе одна внешняя сущность – оператор, которая конвертируется в соответствующую роль. Диаграмма прецедентов для этой роли представлена на рисунке 4.23. С помощью Vpsim.SD нарисована диаграмма последовательности приема документов и ТС (рисунок 4.24).

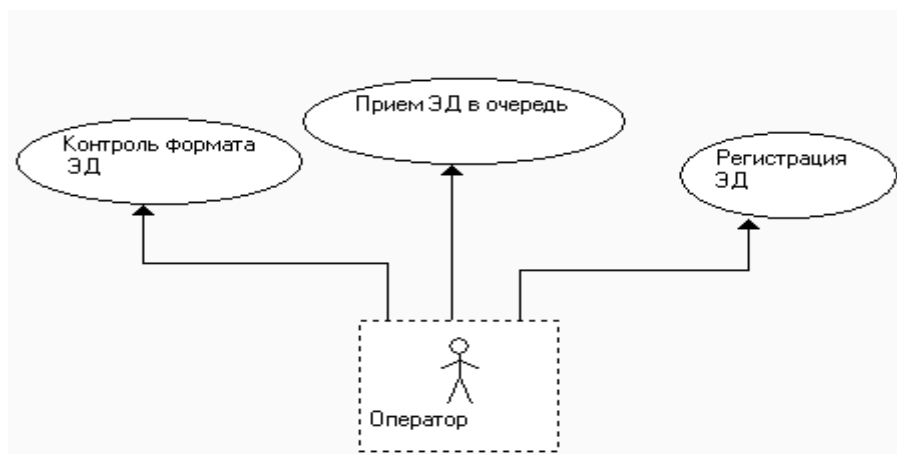


Рисунок 4.23. Диаграмма прецедентов для роли Оператор

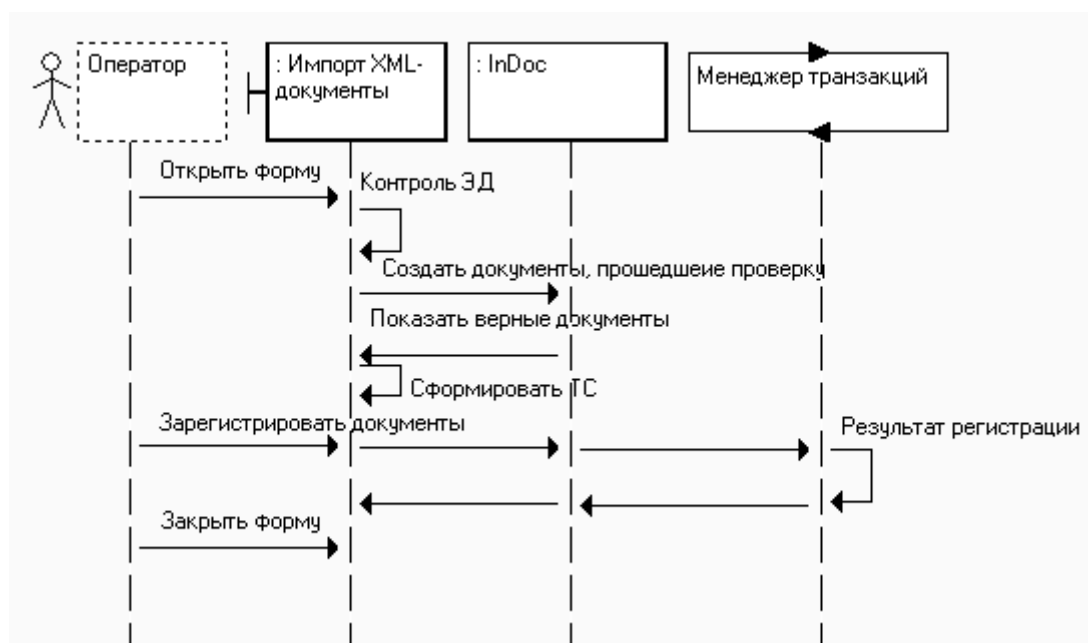


Рисунок 4.24. Диаграмма последовательности приема документов

Формирование ЭД возможно по желанию оператора. Кроме того, есть необходимость массового формирования выписок из реестра на первое и пятнадцатое число каждого месяца, а также справок об операциях на день, в котором были проведены операции. Выполнение этой рутинной задачи было решено возложить на программного агента. DFD-диаграмма процесса формирования ЭД представлена на рисунке 4.25.

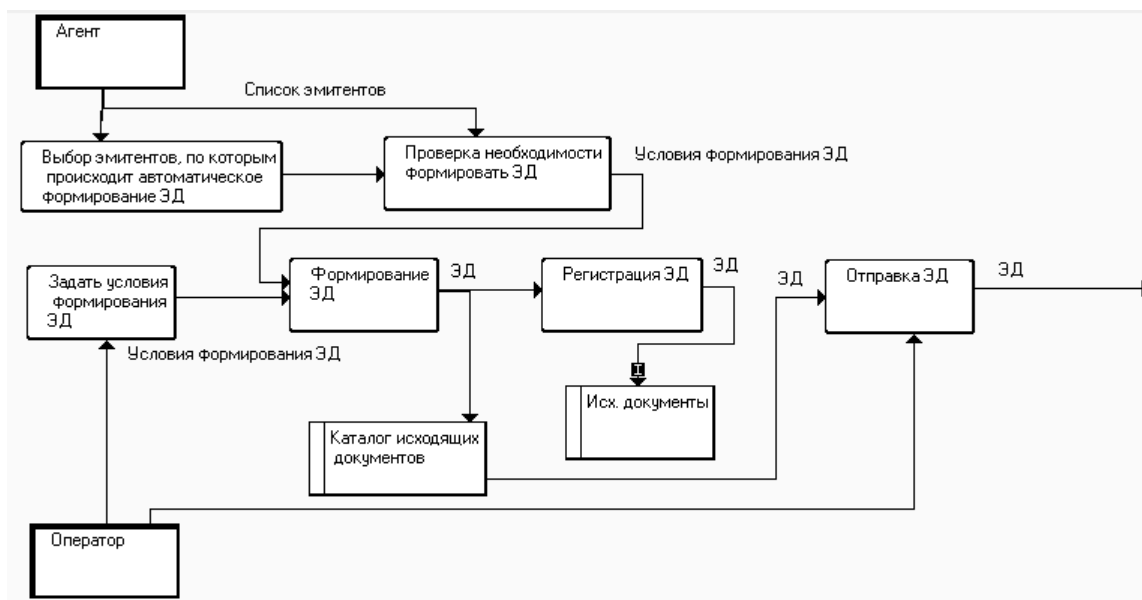


Рисунок 4.25. DFD-диаграмма процесса формирования ЭД

Сценарий поведения агента был представлен в виде диаграммы последовательности (рисунок 4.26).

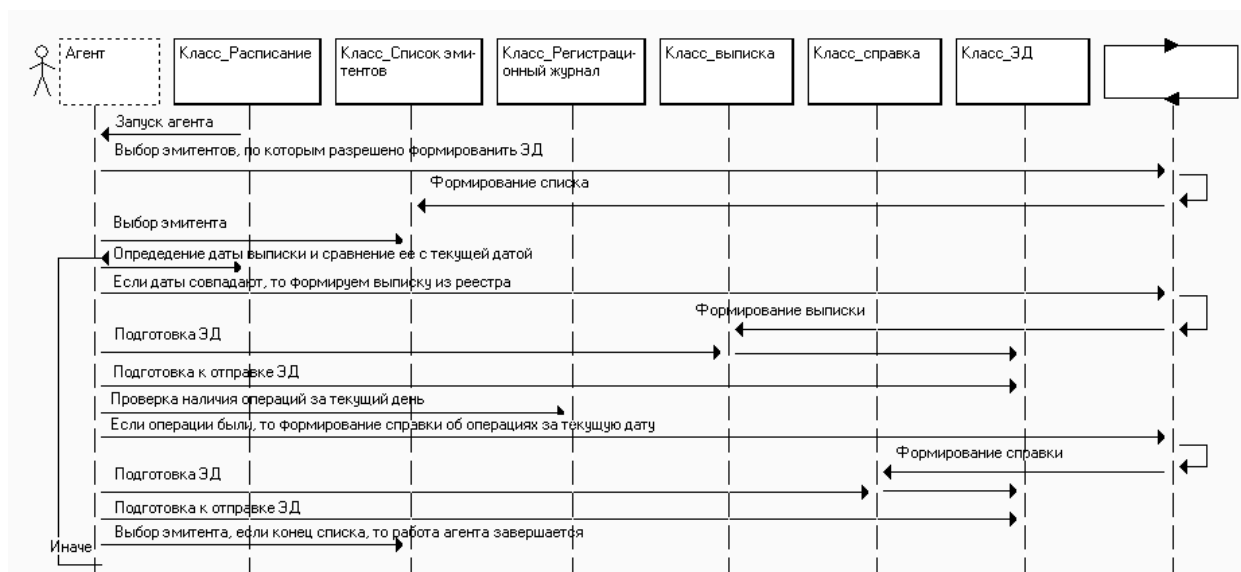


Рисунок 4.26. Сценарий поведения агента

Внедрение электронного документооборота в работе регистратора позволяет существенно сократить трудоемкость приема списков клиентов номинального держателя и ускорить получение им выписок и справок по своим запросам.

В результате проектирования дополнительного модуля и программного агента в Bpsim.SD удалось описать необходимую информацию для их реализации. Предложенный метод и пакет Bpsim позволили упростить

передачу результатов работы между аналитиком и разработчиком ПО за счет преобразования данных модели БП в модель ИС.

4.3. Экспериментальные оценки

Для оценки эффективности предложенного метода и CASE-средства проекты, описанные в главе 3, разработаны с использованием приложений-аналогов (AllFusion, Rational Rose) и BPsim.SD [63; 85]. Весь процесс разработки был разделен на следующие этапы:

1. Построение модели МППР: описание процессов преобразования (преобразователи и ресурсы).
2. Построение модели МППР: описание интеллектуальных агентов.
3. Построение DFD-диаграмм путем конвертации из МППР.
4. Построение диаграммы прецедентов.
5. Построение диаграммы классов: базовые классы.
6. Построение диаграммы классов: другие классы.
7. Наполнение базы знаний экземплярами.
8. Построение диаграммы последовательности.
9. Проектирование структуры БД: проектирование таблиц.
10. Проектирование структуры БД: проектирование связей.
11. Проектирование структуры БД: описание триггеров.
12. Проектирование структуры БД: генерировании БД.
13. Проектирование пользовательского интерфейса.
14. Генерация кода программы: генерация структуры программы.
15. Генерация кода программы: использование алгоритмов агентов в коде.
16. Генерация кода программы: генерация пользовательских интерфейсов.

На рисунке 4.27 показана продолжительность этапов разработки для проекта по анализу БП и разработке технического задания на ЕИС вуза УГТУ-УПИ в системах аналогах и BPsim.SD. Этап 3 в системах аналогах выполнялся вручную, этапы 13, 15-16 в них не выполнялись.

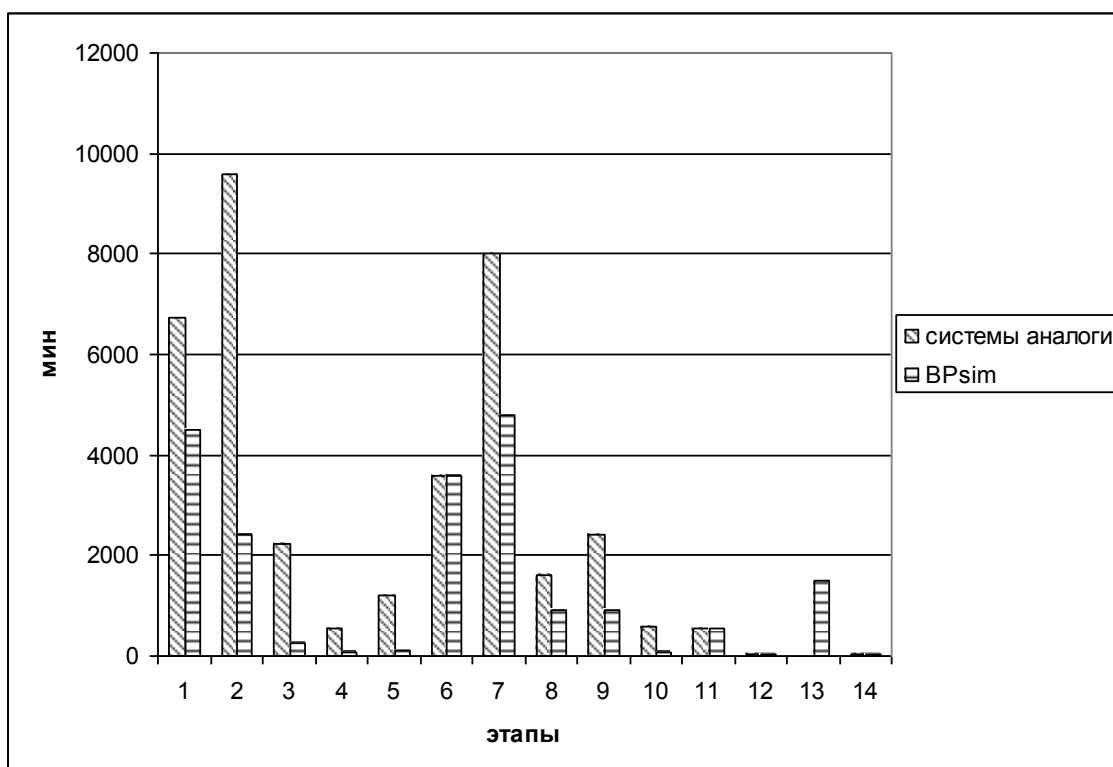


Рисунок 4.27. Продолжительность этапов разработки для проекта по анализу БП и разработке технического задания на ЕИС вуза УГТУ-УПИ

Таблица 4.2

**Оценка эффективности использования BPsim
для проекта ЕИС вуза УГТУ-УПИ**

№ этапа	Продолжительность в системах-аналогах, мин	Продолжительность в BPsim, мин	Эффект, %
1	6750	4500	33
2	9600	2400	75
3	2250	250	89
4	560	70	88
5	1200	100	92
6	3600	3600	0
7	8000	4800	40
8	1620	900	44
9	2400	900	63
10	600	60	90
11	560	560	0
12	20	20	0
13	-	1500	-
14	20	20	0

На рисунке 4.28 показана продолжительность этапов разработки для проекта Модуль «Трансфер-агентский обмен».

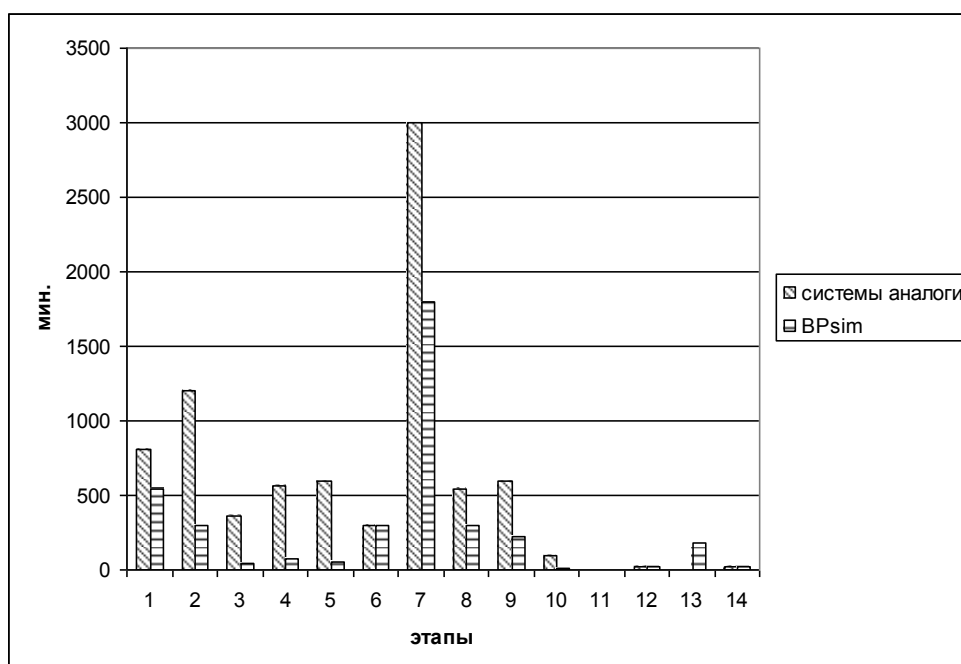


Рисунок 4.28. Продолжительность этапов разработки
для проекта Модуль «Трансфер-агентский обмен»

Таблица 4.3

**Оценка эффективности использования BPsim
для проекта Модуль «Трансфер-агентский обмен»**

№ этапа	Продолжительность в системах-аналогах, мин	Продолжительность в BPsim, мин	Эффект, %
1	810	560	31
2	1200	330	73
3	360	46	87
4	560	70	88
5	600	55	91
6	300	300	0
7	3000	1900	37
8	540	350	35
9	600	275	54
10	100	10	90
11	0	0	0
12	20	20	0
13	-	180	-
14	20	20	0

На рисунке 4.29 показана продолжительность этапов разработки для проекта Модуль «Электронный документооборот».

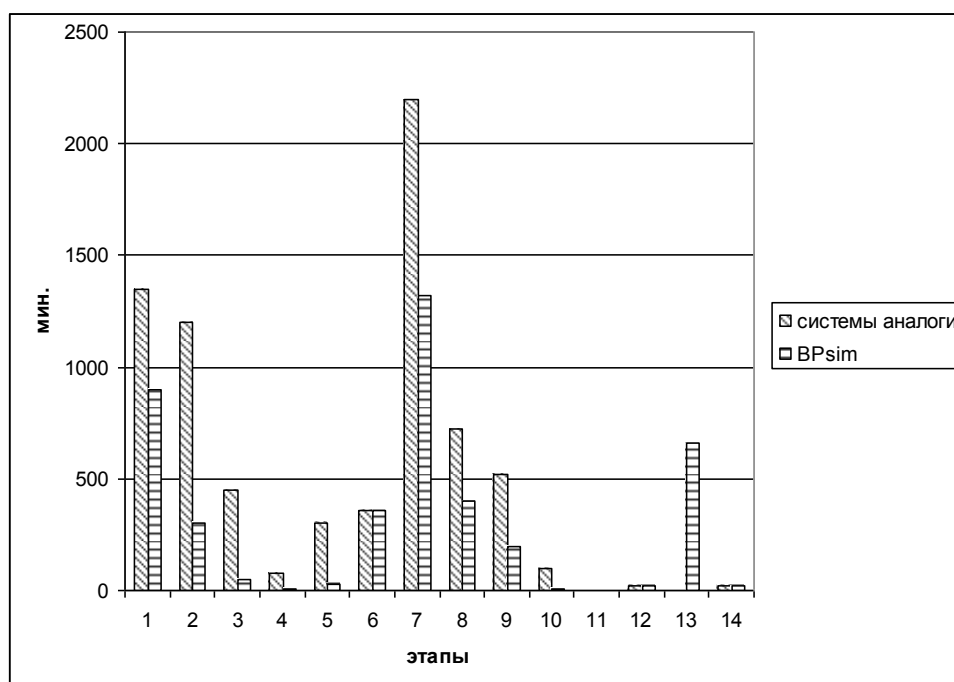


Рисунок 4.29. Продолжительность этапов разработки для проекта Модуль «Электронный документооборот»

Таблица 4.4

Оценка эффективности использования BPsim для проекта Модуль «Электронный документооборот»

№ этапа	Продолжительность в системах-аналогах, мин	Продолжительность в BPsim, мин	Эффект, %
1	1350	910	33
2	1200	335	72
3	450	60	86
4	80	15	81
5	300	22	93
6	360	360	0
7	2200	1340	39
8	720	400	45
9	520	195	63
10	100	10	90
11	0	0	0
12	20	20	0
13	-	660	-
14	20	20	0

На рисунке 4.30 показана продолжительность этапов разработки для проекта Модуль «Входящие документы».

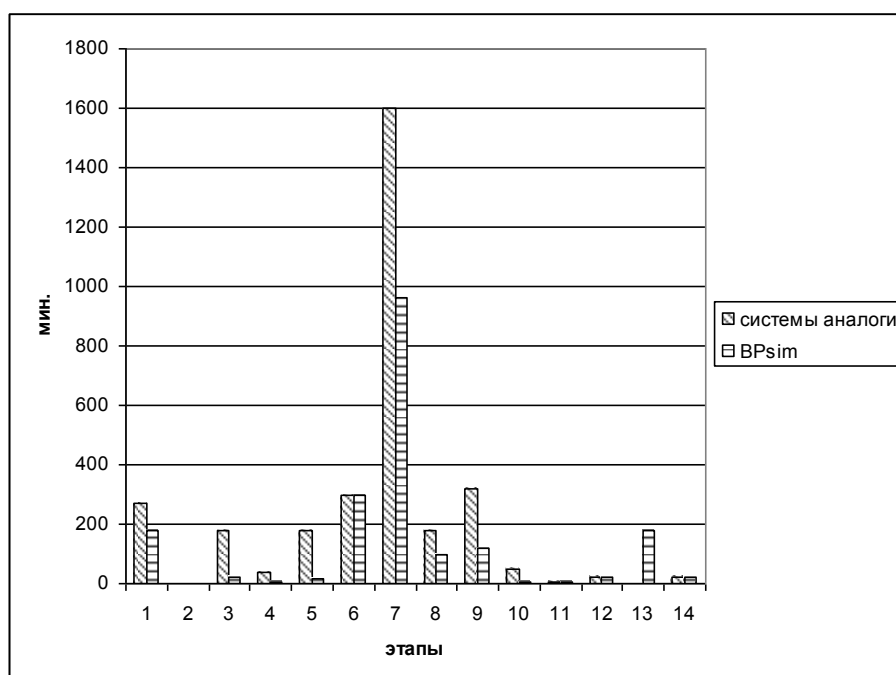


Рисунок 4.30. Продолжительность этапов разработки
для проекта Модуль «Входящие документы»

Таблица 4.5

**Оценка эффективности использования BPsim
для проекта Модуль «Входящие документы»**

№ этапа	Продолжительность в системах-аналогах, мин	Продолжительность в BPsim, мин	Эффект, %
1	280	180	36
2	-	-	-
3	180	20	89
4	40	5	88
5	180	17	91
6	300	300	0
7	1600	960	40
8	180	100	45
9	320	120	62
10	50	5	90
11	5	5	0
12	20	20	0
13	-	180	-
14	20	20	0

Для сравнительного анализа продолжительностей этапов разработки для различных проектов и средств был построен график с логарифмической шкалой по оси Y, приведенный на рисунке 4.31.

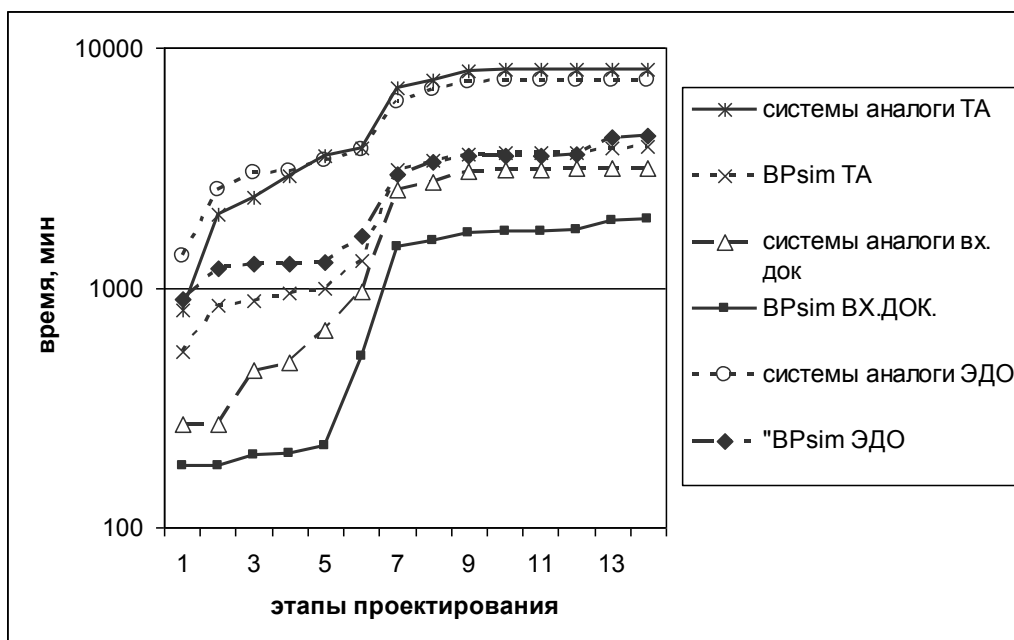


Рисунок 4.31. Продолжительность этапов разработки для всех проектов и средств разработки

Использование BPsim.SD дает следующие преимущества: интеграция бизнес-моделирования, структурного и объектно-ориентированного подхода, конвертация одних диаграмм в другие. Все это приводит к сокращению времени перехода между этапами проекта по сравнению с использованием существующих систем-аналогов. Средний эффект составил 42%.

Выводы по главе 4:

Для оценки эффективности метода ППР в области разработки ИС и пакета BPsim была проведена параллельная разработка ИС с помощью CASE-средств Process Modeler и Rational Rose (система аналог) и BPsim.SD. Для предприятия ЗАО «Ведение реестров компаний» было разработано несколько дополнительных модулей для системы ведения реестров (СВР) «Вереком 2», также был проведен анализ БП и разработка ТЗ на ЕИС вуза УГТУ-УПИ.

ЗАКЛЮЧЕНИЕ

1. Определен перечень характеристик и проведен сравнительный анализ наиболее распространенных CASE-средств: ARIS ToolSet, Power Designer, Borland Together Designer, продукты IBM Rational, CA ERwin Modeling Suite, Biz Agi и Elma. Обследованные системы имеют ряд недостатков: отсутствие интеллектуальности процесса проектирования – не решена задача автоматического перехода к проектированию одних диаграмм на основе других; не позволяют на основе динамической модели процессов ОТС (МППР) построить модель ИС.

2. Разработаны требования к моделям и методу ППР: динамическое моделирование МППР, содержащих модели ИА (ЛПР); ИМ для проверки модели «как будет» на этапе реинжиниринга БП; интеллектуальное проектирование ИС, включающее функциональный и объектно-ориентированный анализ, проектирование ПИ, формирование исполняемого кода ИС и структуры БД.

3. Решена задача системного анализа предметной области ОТС: выявлены три характерных класса процессов: бизнес-процессы, процессы согласования и принятия решений. Определены разрывы в модели знаний, возникающие между пользователем, аналитиком и разработчиком, приводящие к ошибкам при автоматизации ОТС.

4. В результате проведения анализа существующих моделей динамических БП и ППР (сети Петри; расширенные сети Петри; системы массового обслуживания; МППР) и знаний (продукционная модель, семантической сети, фреймовая модель, фреймово-семантическая модель) были выбраны следующие модели. Для решения поставленной задачи в качестве модели ОТС предложено использовать динамическую модель МППР К. А. Аксенова, поскольку она наиболее полно отвечает следующим требованиям: учет временных характеристик, возможность учета различных типов ресурсов, моделирование конфликтов на общих средствах, наличие модели ИА (ЛПР). Выбранная фреймово-семантическая модель представления знаний

А. Н. Швецова имеет следующие преимущества: эффективно реализует иерархическое представление данных, хорошо сочетается с ООП.

5. Разработан метод поддержки принятия решений при разработке ИС предметной области ОТС, который отличается от существующих:

- применением на этапе обследования предметной области мультиагентного имитационного моделирования для анализа процессов ОТС;
- рассмотрением распределенной ИС в виде мультиагентной системы;
- применением фреймово-семантической модели представления знаний на основе фрейм-концептов и концептуальных графов, с целью формализации знаний о предметных областях разработки ИС и ОТС;
- интеграцией структурного и ООП, мультиагентного подхода для решения задачи поддержки принятия решений при автоматизации процессов ОТС;
- преобразованием мультиагентной имитационной модели ОТС в основу модели архитектуры ИС и ее элементов, представление архитектуры в виде структурных диаграмм и диаграмм объектно-ориентированного подхода для обеспечения эффективного взаимодействия между специалистами-предметниками и ИТ-специалистами;
- анализом эффективного распределения БД при наличии риска продолжительных отказов в обслуживании распределенной ИС.

6. На основе моделей и метода ППР в области ИС разработаны:

- интерфейсы СППР, ориентированные на конечного пользователя;
- программное, информационное, алгоритмическое и методическое обеспечение проблемно-ориентированного пакета BPsim.SD;
- технология работы с BPsim.SD.

7. Разработанная СППР BPsim.SD отличается от существующих:

- интеграцией структурного и объектно-ориентированного подходов (ООП) с ИМ МППР;
- возможностью конвертировать структурные диаграммы в диаграммы объектно-ориентированного подхода;
- возможностью создания модели поведения программного агента;

- возможностью создания прототипа форм пользовательского интерфейса непрограммирующим пользователем.

8. Созданный пакет BPsim.SD внедрен в ЗАО «Ведение реестров компаний», в УрФУ (ГОУ ВПО «УГТУ-УПИ»). С его помощью было подготовлено техническое задание на ЕИС вуза, разработаны модули СВР «Вереком» и «Вереком 2». Экономический эффект от внедрения предложенных моделей «как будет» и автоматизации процесса «Движение контингента» составляет 1 027 тыс. руб. в год.

9. Проведен анализ эффективности предложенного метода и реализующего его пакета BPsim.SD. Он показал, что происходит сокращение времени перехода между этапами разработки информационных систем. Средний эффект составил 56% в части разработки модели ИС.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

CASE (Computer Aided Software Engineering) – автоматизированная разработка ПО

UML (Unified Modeling Language) – унифицированный язык моделирования

XP – eXtreme Programming (экстремальное программирование)

БД – база данных

БП – бизнес-процесс

ЕИС – единая информационная система

ЖЦ ПО – жизненный цикл ПО

ИА – интеллектуальные агенты

ИИ – искусственный интеллект

ИМ – имитационное моделирование

ИС – информационная система

ИТ – информационные технологии

КИС – корпоративные информационные системы

ЛПР – лицо, принимающее решение

МАС – мультиагентные системы

МППР – мультиагентные процессы преобразования ресурсов

ООМ – объектно-ориентированные методы

ОТС – организационно-технические системы

ПО – программное обеспечение

ППР – поддержка принятия решения

СВР – система ведения реестров

СМО – системы массового обслуживания

СППР – система поддержки принятия решений

СУБД – система управления базами данных

ТА – трансфер-агент

ТЗ – техническое задание

СПИСОК ЛИТЕРАТУРЫ

1. Аксенов, К. А. Мультиагентное имитационное моделирование бизнес-процессов и организационно-технических систем, среда BPSIM [Текст] / К. А. Аксенов, Е. Ф. Смолий // Материалы 5 Всероссийской научн.-практ. конф. «ИММОД-2011». – СПб. : ЦНИИТС, 2011. – Т. 1. – С. 11-19.
2. Аксенов, К. А. Теория и практика средств поддержки принятия решений [Текст] : монография / К. А. Аксенов. – Germany, Saarbrücken : LAP LAMBERT Academic Publishing GmbH & Co. KG, 2011. – 341 с. – ISBN 978-3-8465-0782-7.
3. Советов, Б. Я. Моделирование систем [Текст] : учеб. для вузов. 3-е изд. / Б. Я. Советов, С. А. Яковлев. – М. : Высш. шк., 2001. – 343 с.
4. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем [Текст] / А. М. Вендров. – М. : Финансы и статистика, 2005. – 544 с.
5. Черемных, С. В. Структурный анализ систем: IDEF-технологии [Текст] / С. В. Черемных, И. О. Семенов, В. С. Ручкин. – М. : Финансы и статистика, 2001. – 208 с.
6. Верников, Г. Основные методологии обследования организаций. Стандарт IDEF0 [Электронный ресурс] / Г. Верников. – Режим доступа: <http://www.interface.ru>.
7. Калашян, А. Н. Структурные модели бизнеса: DFD-технологии [Текст] / А. Н. Калашян, Г. Н. Калянов. – М. : Финансы и статистика, 2009. – 256 с.
8. Stephen, A. White, Derek Miers BPMN Modeling and Reference Guide [Text] / A. Stephen. – Future Strategies Inc., 2008. – 226 p.
9. Верников, Г. Стандарт онтологического исследования IDEF5 [Электронный ресурс] / Г. Верников. – Режим доступа: <http://vernikov.ru/krisis/item/31--idef5.html>.
10. Грошев, А. С. Информатика [Текст] : учебник для вузов / А. С. Грошев. – Архангельск : Арханг. гос. техн. ун-т, 2010. – 470 с.

11. Фаулер, М. UML в кратком изложении: Применение стандартного языка объектного моделирования [Текст] : пер. с англ. / М. Фаулер, К. Скотт. – М. : Мир, 1999.
12. UML – The Unified Modeling Language [Электронный ресурс]. – Режим доступа: <http://www.uml.org>.
13. Леоненков, А. Самоучитель UML 2 [Текст] / А. Леоненков. – СПб. : БХВ-Петербург, 2007. – 558 с.
14. Хаммер, М. Реинжиниринг корпорации (манифест революции в бизнесе) [Текст] / М. Хаммер, Дж. Чампи. – СПб. : Санкт-Петербургский университет, 1997. – 332 с.
15. Смирнова, Г. Н. Проектирование экономических информационных систем [Текст] / Г. Н. Смирнова, А. А. Сорокин, Ю. Ф. Тельнов. – М. : Финансы и статистика, 2001. – 510 с.
16. Имитационное моделирование производственных систем [Текст] / под общ. ред. А. А. Вавилова. – М. : Машиностроение ; Берлин : Техник, 1983. – 416 с.
17. Аксенов, К. А. Исследование и разработка средств имитационного моделирования дискретных процессов преобразования ресурсов [Текст] : дис. ... канд. техн. наук : 05.13.18 / Аксенов К. А. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2003. – 188 с.
18. Технология системного моделирования [Текст] / Е. Ф. Аврамчук [и др.] ; под общ. ред. С. В. Емельянова. – М. : Машиностроение ; Берлин : Техник, 1988. – 520 с.
19. Рыжиков, Ю. И. Имитационное моделирование: Теория и технологии [Текст] / Ю. И. Рыжиков. – М. : Альтекс-А, 2004. – 384 с.
20. Бек, К. Экстремальное программирование [Текст] / К. Бек. – СПб. : Питер, 2002.
21. Colin, J. Neill and Phillip A. Laplante Requirements Engineering: The State of the Practice [Text] / J. Colin // m-n IEEE Software. – 2003, November/December. – P. 40-45.

- 22.Щенников, С. Ю. Реинжиниринг бизнес-процессов. Экспертное моделирование, управление, планирование и оценка [Текст] / С. Ю. Щенников. – Ось-89, 2004. – 288 с.
- 23.Куприянов, В. В. САПР и системы искусственного интеллекта на базе ЭВМ [Текст] / В. В. Куприянов [и др.]. – М. : Наука, 1991. – 159 с.
- 24.Андрейчиков, А. В. Интеллектуальные информационные системы [Текст] / А. В. Андрейчиков, О. Н. Андрейчикова. – М. : Финансы и статистика, 2004. – 422 с.
- 25.Поспелов, Д. А. Многоагентные системы – настоящее и будущее [Текст] / Д. А. Поспелов // Информационные технологии и вычислительные системы. – 1998. – № 1.
- 26.Wooldridge, M. Intelligent Agents: Theory and Practice [Text] / M. Wooldridge, N. Jennings // The Knowledge Engineering Review. – 1995. – Vol. 10. – № 2. – P. 115-152.
- 27.Вендров, А. М. CASE-технологии. Современные методы и средства проектирования информационных систем [Текст] / А. М. Вендров. – М. : Финансы и статистика, 1998.
- 28.Маклаков, С. Объединение структурного и объектного подхода в новом поколении CASE-средств Computer Associates [Электронный ресурс] / С. Маклаков. – 2001. – Режим доступа <http://www.interface.ru>.
- 29.Гамильтон Гайес (Hamilton Hayes), менеджер линейки продуктов AllFusion Переведено БНТП по заказу Interface Ltd. Ценность моделирования и анализа бизнес-процессов при помощи линейки продуктов AllFusion от компании Computer Associates [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/fset.asp?Url=/ca/cennmodallfca.htm>.
- 30.CA ERwin Modeling Suite 7.3 [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/home.asp?artId=7561>.
- 31.Функциональные возможности AllFusion Process Modeler 7 (BPwin) [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/home.asp?artId=5819>.

32.AllFusion Component Modeler 5.0 [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/home.asp?artId=132>.

33.Новичков, А. Эффективная разработка программного обеспечения с использованием технологий и инструментов компании RATIONAL [Электронный ресурс] / А. Новичков. – 2000. – Режим доступа: <http://www.interface.ru>.

34.Джеймс Рамбо, специалист по методологии, компания Rational Software Тенденции в развитии языка UML и разработки ПО [Электронный ресурс]. – Режим доступа: http://www.business-process.ru/designing/methodology/uml/article/tenden_uml_article.html.

35.IBM Rational [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/home.asp?artId=62&vId=19>.

36.Арнольд, С. Что нового в IBM Rational Software Architect 8.0 [Электронный ресурс] / С. Арнольд. – Режим доступа: <http://www.ibm.com/developerworks/ru/library/r-whats-new-in-rational-software-architect-8>.

37.Ивлев, В. Применение ARIS Toolset для сертификации по международным стандартам серии ИСО 9000 [Электронный ресурс] / В. Ивлев, Т. Попова. – Режим доступа: http://consulting.ru/econs_wp_4248.

38.Вышла новая версия решения для анализа бизнес-процессов Oracle BPA Suite 11g [Электронный ресурс]. – Режим доступа: <http://www.olap.ru/home.asp?artId=826>.

39.Sybase PowerDesigner 12 [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/home.asp?artId=3952>.

40.Together 2008 [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/home.asp?artId=18125>.

41.Bigazi [Электронный ресурс]. – Режим доступа: <http://www.slideshare.net/gibtech/bizagi-bpm>.

42.Бизнес консоль Bigazi [Электронный ресурс]. – Режим доступа: <http://www.b-k.ru/products/bizagi/>.

43.Пакет ELMA BPM Suite [Электронный ресурс]. – Режим доступа: <http://crm74.ru/page/elma-bpm-suite>.

44.Аксенов, К. А. Исследование и разработка средств имитационного моделирования дискретных процессов преобразования ресурсов [Электронный ресурс] : дис. ... канд. техн. наук : 05.13.18 / Аксенов К. А. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2003. – 188 с.

45.ОТЧЕТ о выполнении НИОКР по теме: «Разработка моделей, методов и системы динамического моделирования ситуаций и автоматизации проектирования программного обеспечения в области процессов преобразования ресурсов» (государственный контракт № 5058р/7296 от 03.04.2007) Этап № 2.1 «Разработка методов, моделей и алгоритмов проектирования информационных систем для предметной области процессов преобразования ресурсов» (промежуточный) Березовский 2007.

46.Гончарова, Н. В. Разработка и применение системы динамического моделирования мультиагентных процессов преобразования ресурсов [Электронный ресурс] : дис. ... канд. техн. наук : 05.13.18 / Гончарова Н. В. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2006. – 247 с.

47.Аксенов, К. А. Динамическое моделирование мультиагентных процессов преобразования ресурсов [Текст] / К. А. Аксенов, Н. В. Гончарова. – Екатеринбург : ГОУ ВПО «Уральский государственный технический университет – УПИ», 2006.

48.Джексон, П. Введение в экспертные системы [Текст] / П. Джексон. – М. – СПб. – Киев : Вильямс, 2001 – 622 с.

49.Казиев, В. М. Введение в анализ, синтез и моделирование систем БИНОМ [Текст] / В. М. Казиев. – Лаборатория знаний, Интернет-университет информационных технологий – ИНТУИТ.ру, 2006.

50.Minsky, M. A framework for Representing Knowledge in The Psychology of Computer Vision [Text] / M. Minsky, P. H. Winston (ed.). – McGraw-Hill, 1975.

51. Швецов, А. Н. Модели и методы построения корпоративных интеллектуальных систем поддержки принятия решений [Текст] : дис. ... д-ра техн. наук : 05.13.01 / Швецов А. Н. – Санкт-Петербург, 2004. – 461 с.

52. Sowa, J. F. Conceptual graphs for a database interface [Text] / J. F. Sowa // IBM Journal of Research and Development. – 1976. – № 20:4. – P. 336-357.

53. Sowa, J. F. Conceptual Structures : Information Processing in Mind and Machine [Text] / J. F. Sowa. – Reading, MA : Addison – Wesley, 1984. – 481 p.

54. Sowa, J. F. Knowledge Representation : Logical, Philosophical, and Computational Foundations [Text] / J. F. Sowa. – Pacific Grove, CA : Brooks / Cole Publishing Co., 2000. – 594 p.

55. Aksyonov, K. A. Enterprise Information Systems Engineering Method Based on Semantic Models of Multi-Agent Resource Conversion Processes and Software [Text] / K. A. Aksyonov, I. A. Spitsina, E. A. Bykov, N. V. Goncharova // IADIS Multi Conference on Computer Science and Information Systems – MCCSIS'08 : International conference Intelligent Systems and Agents 2008 part of the IADIS Multi Conference on Computer Science and Information Systems 2008, Amsterdam, The Netherlands, July 22-24, 2008. – Proceedings of the IADIS International conference Intelligent Systems and Agents 2008, July 2008. – P. 225-227.

56. Разработка методов, моделей и алгоритмов проектирования информационных систем для предметной области процессов преобразования ресурсов [Текст] : Отчет по проекту № 01.2.007 08048- / ООО «НПП «Системы автоматизации поддержки бизнеса» ; руковод. работы К. А. Аксенов ; исполн. И. А. Спицина, Е. Ф. Смолий, О. П. Аксенова и [др.]. – Березовский, 2007. – 55 с.

57. Спицина, И. А. Особенности методологии проектирования информационных систем для предметной области мультиагентных процессов преобразования ресурсов [Текст] / И. А. Спицина, К. А. Аксенов // Журнал научных публикаций аспирантов и докторантов. – 2008. – № 8.

58. Аксенов, К. А. Анализ и синтез процессов преобразования ресурсов на основе имитационного моделирования и интеллектуальных агентов [Текст] / К. А. Аксенов, А. С. Антонова, И. А. Спицина // Научно-технические ведомости

СПбГПУ. Информатика. Телекоммуникации. Управление. – 2011. – № 1 (115). – С. 13-20.

59.Пищулов, В. Введение в теорию производства [Текст] : учеб. пособие / В. Пищулов, К. Рихтер, Е. Дятел. – Екатеринбург : Изд-во Урал. гос. экон. ун-та, 2003. – 161 с.

60.Аксенов, К. А. Использование онтологии для автоматизации разработки программного обеспечения и CASE-средство «BPsim.SD» [Текст] / К. А. Аксенов, И. А. Спицина // Пятая Международн. научно-практ. конф. «Теория, методы проектирования, программно-техническая платформа корпоративных информационных систем». – Новочеркасск, 2007. – С. 228-230.

61.Аксенов, К. А. Проектирование АСУ предприятия на основе семантических моделей мультиагентного процесса преобразования ресурсов и программного обеспечения [Текст] / К. А. Аксенов, И. А. Спицина // Информационно-математические технологии в экономике, технике и образовании : сборники тезисов Второй Международной научной конференции. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2007. – С. 70-72.

62.Спицина, И. А. Семантическая модель информационной системы и автоматизация учебного процесса [Текст] / И. А. Спицина // Научные труды международной научно-практической конференции «СВЯЗЬ-ПРОМ 2008» в рамках 5го Евро-Азиатского форума «СВЯЗЬ-ПРОМЭКСПО 2008». – Екатеринбург : ЗАО «Компания Реал-Медиа», 2008. – С. 293.

63.Аксенов, К. А. Оценка эффективности метода поддержки принятия решений разработки информационных систем предметной области мультиагентных процессов преобразования ресурсов [Текст] / К. А. Аксенов, И. А. Спицина // Информационно-математические технологии в экономике, технике и образовании : тезисы докладов 3-й Международной научной конференции. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2008. – С. 158-159.

64.Исследование операций [Текст] : в 2 т. : пер. с англ. / под ред. Дж. Моудера, С. Элмаграби. – М. : Мир, 1981.

65. Анфилатов, В. С. Системный анализ в управлении [Текст] / В. С. Анфилатов, А. А. Емельянов, А. А. Кукушкин. – М. : Финансы и статистика, 2002. – 368 с.
66. IBM, Dell, Fujitsu & Stratus Get Highest Marks in ITIC Reliability Survey [Электронный ресурс]. – Режим доступа <http://itic-corp.com/blog/2013/02/ibm-dell-fujitsu-stratus-get-highest-marks-in-itic-reliability-survey/#more-1021>.
67. Бройдо, В. Л. Вычислительные системы, сети и телекоммуникации [Текст] / В. Л. Бройдо. – СПб. : Изд-во «Питер», 2004.
68. Mathworld.Wolfram [Электронный ресурс]. – Режим доступа: <http://mathworld.wolfram.com/Run.html>.
69. Introduction to mathematical probability [Text] / J. V. Uspensky, Mc-Groqhll. – NY–London, 1937.
70. Tong, Y. L. A rearrangement inequality for the longest run with an application to network reliability [Text] / Y. L. Tong // Journal of Applied Probability. – 1985. – № 22. – P. 386-393.
71. Section 14.1 Problems and Snapshots from the World of Probability by Blom, Holst, and Sandell.
72. Скобелев, О. П. Открытые мультиагентные системы для оперативной обработки информации в процессах принятия решений [Текст] : автореф. дис. ... д-ра техн. наук / Скобелев О. П. – Самара, 2003. – 35 с.
73. Андреев, В. В. Инструментальные средства для разработки мультиагентных систем промышленного масштаба [Электронный ресурс] / В. В. Андреев [и др.]. – Режим доступа: http://www.ssc.smr.ru/media/ipuss_conf/06/5_01.pdf.
74. Zambonelli, F. Developing Multiagent systems: The GAIA methodology [Text] / F. Zambonelli, N. Jennings, M. Wooldridge // ACM Transactions on Software Engineering and Methodology. – 2003. – № 12 (3). P. 417-470.
75. Gorodetsky, V. Support for Analysis, Design and Implementation Stages with MASDK [Text] / V. Gorodetsky, O. Karsaev, V. Samoylov, V. Konushy // LNCS. – 2009. – № 5386. – P. 272-287.

76. Городецкий, В. И. Технология разработки прикладных многоагентных систем в инструментальной среде MASDK [Текст] / В. И. Городецкий [и др.] // Труды СПИИРАН. – 2006. – Вып. 3. – Т. 1. – ISSN 2078-9181 (печ.) ; ISSN 2078-9599 (онлайн).

77. Швецов, А. Н. Проектирование прикладных мультиагентных систем с использованием пакета DISIT [Текст] / А. Н. Швецов, М. А. Сергушичева // Информационные технологии. – 2009. – № 8. – С. 54-60.

78. Александров, Д. В. Консалтинг при информатизации организаций [Текст] : учеб. пособ. / Д. В. Александров, Д. Н. Фадин ; Владим. гос. ун-т. – Владимир : Изд-во Владим. гос. ун-та, 2006. – 72 с.

79. Спицина, И. А. Численный анализ задержек в синхронной информационной системе [Электронный ресурс] / И. А. Спицина, А. Л. Крохин // Международная научно-техническая интернет-конференция «Информационные системы и технологии». – 2013. – Режим доступа: <http://isit-conf.gu-unpk.ru/conferences/2/materials/index/415>.

80. Вентцель, Е. С. Теория вероятностей и ее инженерные приложения [Текст] / Е. С. Вентцель, Л. А. Овчаров. – М. : Наука. Главная редакция физико-математической литературы, 1988. – 480 с.

81. Dorokhov, A. Simulation simple models and comparison with queuing theory [Электронный ресурс] / A. Dorokhov. – Режим доступа: Monarc.web.cern.ch/monarc/docs/monarc_docs/.

82. Аксенов, К. А. Разработка CASE-средства на основе интеграции функционального и объектно-ориентированного моделирования [Текст] / К. А. Аксенов [и др.] // X отчетная конференция молодых ученых ГОУ ВПО УГТУ-УПИ. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2006. – С. 285-288.

83. Аксенов, К. А. CASE-средство Software Engineer – объединение структурного и объектно-ориентированных подходов [Текст] / К. А. Аксенов [и др.] // Региональная научно-практическая конференция «Методы разработки

программного обеспечения». – Екатеринбург, 2006. – Режим доступа: <http://webconf.rtf.ustu.ru/>.

84. Аксенов, К. А. CASE-средство BPsim.SD [Текст] / К. А. Аксенов, И. А. Спицина // XII отчетная конференция молодых ученых ГОУ ВПО УГТУ-УПИ. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2007. – С. 252-257.

85. Аксенов, К. А. Разработка и применение метода поддержки принятия решений в области создания информационных систем [Текст] / К. А. Аксенов, И. А. Спицина // Научные труды международной научно-практической конференции «СВЯЗЬ-ПРОМ 2009» в рамках 5-го Евро-Азиатского форума «СВЯЗЬ-ПРОМЭКСПО 2009». – Екатеринбург : ЗАО «Компания Реал-Медиа», 2009. – С. 133-135.

86. Спицина, И. А. Опыт использования CASE-средств при проектировании информационных систем (ИС) [Текст] / И. А. Спицина, К. А. Аксенов // Информационно-математические технологии в экономике, технике и образовании : сборники материалов Областной научно-практической конференции. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2006. – Ч. 1. – С. 187-193.

87. Результаты обследования и формирования требований на создание единой ИС поддержки учебного процесса (с 15 мая 2005 года по 14 февраля 2006 года) [Текст] : Отчет по проекту № *01200601073 / ГОУ ВПО «Уральский государственный технический университет – УПИ» ; руковод. работы А. К. Аксенов ; исполн. А. М. Сысков, О. П. Аксенова, А. А. Баронихина, И. А. Спицина [и др.]. – Екатеринбург, 2006. – 119 с.

88. Экспериментальные исследования и верификация применимости информационных систем, методик и моделей на примере создания ИС УГТУ-УПИ «Учеба» [Текст] : Отчет по проекту № *01200601073 / ГОУ ВПО «Уральский государственный технический университет – УПИ» ; руковод. работы О. И. Ребрин ; исполн. И. И. Шолина, А. М. Сысков, К. А. Аксенов, Р. М. Кадушников, И. А. Спицина [и др.]. – Екатеринбург, 2006. – 20 с.

89. Фурсенко, О. В. Представление учебного процесса в виде мультиагентной системы [Текст] / О. В. Фурсенко [и др.] // X отчетная конференция молодых ученых ГОУ ВПО УГТУ-УПИ. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2006. – С. 260-262.

90. Аксенов, К. А. Использование CASE-средств при проектировании программного обеспечения единой информационной системы учебного процесса [Текст] / К. А. Аксенов [и др.] // X отчетная конференция молодых ученых ГОУ ВПО УГТУ-УПИ. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2006. – С. 282-284.

91. Спицина, И. А. Применение методологии UML при проектировании ИС «Электронный деканат» [Текст] / И. А. Спицина, О. В. Фурсенко, К. А. Аксенов // Международная НПК «Новые образовательные технологии в вузе» НОТВ-2005. – Екатеринбург, 2005. – С. 277-278.

92. Фурсенко, О. В. Использование CASE-средств при проектировании информационной системы «Электронный деканат» [Текст] / О. В. Фурсенко [и др.] // IX отчетная конференция молодых ученых ГОУ ВПО УГТУ-УПИ. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2005. – С. 164-171.

93. Аксенов, К. А. Представление учебного процесса в виде мультиагентной системы [Электронный ресурс] / К. А. Аксенов [и др.] // Десятая Всероссийская студенческая научно-техническая Интернет-конференция «Информационные технологии и электроника». – Екатеринбург, 2005. – Режим доступа: <http://webconf.rtf.ustu.ru>.

94. Аксенов, К. А. Разработка системы поддержки принятия решений руководителя вуза в сфере управления проектами информатизации [Текст] / К. А. Аксенов [и др.] // VI-я научно-техническая конференция «Научное программное обеспечение в образовании и научных исследованиях» в составе международной конференции МГУ «Современные информационные технологии и ИТ-образование». – Санкт-Петербург, 2008. – С. 230-235.

95. Техническое задание на создание единой информационной системы учебного процесса УГТУ-УПИ [Текст] : Отчет по проекту № *01200601073 /

ГОУ ВПО «Уральский государственный технический университет – УПИ» ;
руковод. работы А. К. Аксенов, А. Лукшин ; исполн. А. М. Сысков, О. П.
Аксенова, М. В. Марков, И. А. Спицина [и др.]. – Екатеринбург, 2006. – 116 с.

96. Аксенов, К. А., Использование интеллектуального case-средства при автоматизации учебного процесса [Текст] / К. А. Аксенов, И. А. Спицина // VI-я научно-техническая конференция «Научное программное обеспечение в образовании и научных исследованиях» в составе международной конференции МГУ «Современные информационные технологии и ИТ-образование». – Санкт-Петербург, 2008. – С. 236-239.

97. «Форматы электронного взаимодействия регистраторов и депозитариев», утвержденные Советом директоров (протокол No 02/2007 от 30.03.2007г.) [Электронный ресурс]. – Режим доступа: http://www.partad.ru/po/index_docum.html.

Научное издание

**Метод поддержки принятия решений
при разработке информационных систем
на основе мультиагентного подхода**

Редакционно-издательская обработка: Адясова О.А.

Подписано в печать 04.10.2018. Формат 60x84 ¹/₁₆.
Бумага для множ. аппаратов. Печать на ризографе.
Гарнитура «Times New Roman».
Усл. печ. л. 9. Уч.-изд. л. 4,6.
Тираж 500. Заказ.

Оригинал-макет отпечатан в отделе множительной техники
Уральского государственного педагогического университета.
620017 Екатеринбург, пр-т Космонавтов, 26.
E-mail: uspu@uspu.me